MOLTA PS3 "AL FUOCO"

Si dice di avere molta carne al fuoco quando si fa riferimento ad un accadimento di cui si hanno molte notizie e fonti (a dire il vero non solo per quello). L'esempio calza perfettamente per quanto riguarda la PS3.

I lettori più attenti avranno notato che nel corso dell'anno abbiamo trattato più volte questo argomento, prima con l'exploit di "GeoHoz", poi con la notizia della modifica hardware con chiavetta USB che consente di caricare giochi copiati e, infine, con un articolo, nel numero scorso, in cui si tracciava un po' il punto delle varie modifiche hardware in circolazione.

L'articolo "PS3 bucata", di questo numero, fa un po' il punto di tutto quanto è successo in tempi più o meno recenti e su quanto ci dobbiamo aspettare per l'immediato futuro. E' una sorta di documento riassuntivo ma non finale.

Infatti ci aspetta ancora la "prova su strada" della chiave USB per sbloccare la PS3, che abbiamo ordinato via internet e che stiamo testando. Vi forniremo, probabilmente sul prossimo numero, una recensione dettagliata con tutte le nostre impressioni.

Al di là del gioco, che peraltro è una cosa serissima, questo numero segnala un po' un ritorno alle origini, con molti articoli dedicati proprio all'hacking, la vera essenza della nostra rivista.

Come sempre: buona lettura!

Altair

laboratorio@hackerjournal.it
Questo indirizzo è stato creato per inviare articoli, codici, spunti e idee. E' quindi proprio una sorta di "incubatore di idee".

posta@hackerjournal.it
E' l'account creato per l'omonima rubrica che è ricomparsa nelle pagine della rivista. A questo indirizzo dovete inviare tutte le mail che volete vengano pubblicate su HJ.

redazione@hackerjournal.it
Questo è l'indirizzo canonico. Quello con cui potete avere un filo diretto, sempre, con la redazione, per qualsiasi motivo che non rientri nelle due precedenti categorie di posta.

Sommario

Anno 10 - N.207
Novembre 2010

Editore (sede legale)
WLF Publishing S.r.l.
Socio Unico Medi & Son S.r.l.
via Donatello 71 - 00196 Roma
Fax 063214606

Realizzazione editoriale
Progetti e promozioni Srl
redazione@progettiepromozioni.com

Printing
Arti Grafiche Boccia S.p.A. - Salerno

Distributore
M-DIS Distributore SPA
Via Cazzaniga 19 - 20132 Milano

Hacker Journal
Pubblicazione registrata al Tribunale di Milano il 27/10/03 con il numero 601.
Una copia: 2,00 euro

Direttore Responsabile
Teresa Carsaniga
redazione@hackerjournal.it

Direttore Editoriale
Andrea Franchini

Le donne
LO FANNO MEGLIO

Inutile che scambiate sguardi maliziosi con la signorina casualmente seduta di fianco a voi, magari sulla metropolitana o sul tram, il riferimento è alle abitudini informatiche. Un'interessante indagine pubblicata da Symantec infatti sottolinea come le donne abbiano un approccio alla rete decisamente più consapevole in tema di sicurezza informatica. Dall'indagine, infatti, gli uomini si dimostrano più sprovveduti delle donne nei comportamenti online, lasciandosi sedurre più facilmente da e-mail dal contenuto erotico o che promettono denaro facile che si rivelano poi delle truffe. Inoltre, sono più propensi a condividere, anche con estranei, informazioni riservate: il 62% (contro il 65% delle donne) evita di lasciare il proprio numero di carta di credito su Internet.
Quando si parla di shopping, però, si riscattano: sono molto più prudenti infatti rispetto al gentil sesso, dichiarando di utilizzare diverse carte di credito e indirizzi e-mail per effettuare acquisti online in totale sicurezza. Pur di riuscire ad acquistare le scarpe alla moda tanto sognate, le donne si dimostrano invece troppo frettolose nello sfoderare la propria carta di credito e rilasciare informazioni sul proprio conto corrente. Un dato sorprendente e inaspettato riguarda l'utilizzo di Internet per spettegolare: è emerso infatti che fare gossip online agli uomini piaccia più delle donne, che invece si dimostrano più riservate. Inoltre, mentre il 29% degli uomini pubblica in rete foto imbarazzanti degli amici, il 51% delle donne chiede addirittura il permesso prima di taggare sui social network.
Credete che sia finita qui? Non proprio. Infatti si aggiungono altre note che fanno lievitare la propensione alla sicurezza del sesso debole. Le donne non ripetono lo stesso errore due volte: il 54% (contro il 48% degli uomini) dichiara di aver modificato le proprie abitudini per evitare altre esperienze negative, mentre le meno tecnologiche chiedono aiuto ad amici, familiari o esperti per risolvere il problema. Gli uomini invece tendono a sottovalutare l'accaduto, senza preoccuparsene troppo: solo il 29% (contro il 34% femminile) si rassegna a cambiare le proprie abitudini di navigazione, evitando di visitare i siti più rischiosi.
Infine, il luogo comune che vorrebbe gli uomini più tecnologici rispetto alle donne sembra avere radici fondate: dall'indagine emerge infatti che sono più propensi ad usare alcuni accorgimenti per la propria sicurezza online, come tenere aggiornato il software di protezione (69%) ed effettuare regolarmente il backup dei dati (33%).

KASPERSKY LAB E MICROSOFT COOPERANO PER RISOLVERE LA VULNERABILITÀ "NEW ZERO-DAY" SFRUTTATA DAL WORM STUXNET

Kaspersky Lab annuncia di aver collaborato con Microsoft per risolvere con successo una grave vulnerabilità nel sistema Microsoft Windows.
La vulnerabilità è stata classificata nel tipo "zero-day" quando è stata rilevata, ed è stata usata dal famigerato worm Stuxnet. Il Worm.Win32.Stuxnet è noto fondamentalmente come uno strumento di spionaggio industriale: è stato infatti progettato per avere accesso al sistema operativo Siemens WinCC, utilizzato per la raccolta dei dati e per il monitoraggio della produzione. Fin dalla sua prima comparsa nel luglio del 2010, gli specialisti di sicurezza IT hanno analizzato approfonditamente Worm.Win32.Stuxnet. Gli esperti di Kaspersky Lab hanno svolto numerose ricerche per conoscere le caratteristiche di Stuxnet scoprendo che, oltre alla vulnerabilità nel trattamento di file LNK e PIF rilevata inizialmente, il worm utilizza anche altre quattro vulnerabilità presenti in Windows.
Un esempio è l'MS08-067, che è stato usato anche dal famigerato worm Kido (Conficker) nei primi mesi del 2009. Le altre tre vulnerabilità erano precedentemente sconosciute e sono presenti nella versione attuale di Windows.
Insieme a l'MS08-067, Stuxnet utilizza anche un'altra vulnerabilità per diffondersi. Questa è presente nel servizio di stampa Windows Print Spooler, e può essere utilizzata per inviare un codice maligno a un computer remoto, dove poi viene eseguito. In virtù delle caratteristiche di questa vulnerabilità, l'infezione può diffondersi nei computer utilizzando una stampante o tramite l'accesso condiviso ad una di queste. Dopo aver infettato un computer connesso ad una rete, Stuxnet tenta quindi di diffondersi sugli altri computer.
La vulnerabilità è stata classificata come "Print Spooler Service Impersonation" ed è stata valutata come "critica".
Microsoft ha iniziato a lavorare immediatamente per riparare la vulnerabilità ed ha successivamente rilasciato la patch MS10-061, il 14 settembre 2010.
Gli esperti di Kaspersky Lab hanno poi rilevato un'altra vulnerabilità zero-day nel codice Stuxnet.
E' stata classificata come una vulnerabilità "Elevation of Privilege" (EOP), che potrebbe essere sfruttata dal worm per ottenere il controllo completo sul computer infetto. Una vulnerabilità simile di tipo EOP è stata rilevata dagli esperti di Microsoft. Entrambe le vulnerabilità verranno corrette nei prossimi aggiornamenti di sicurezza per sistemi operativi Windows.

Microsoft®

NESSUNA CHIRURGIA AL LASER PUÒ CANCELLARE UN TATUAGGIO DIGITALE

A chi non è mai capitato di essere taggato senza permesso dagli amici in foto imbarazzanti, commentate magari da post altrettanto fastidiosi? L'incubo di essere screditati in rete accomuna la maggior parte degli intervistati, infatti il 45% arriva a pensare che sia impossibile ripristinare completamente la propria reputazione online. I più pessimisti sono i canadesi (57%), mentre gli italiani si collocano in una posizione intermedia (42%). L'ottimismo invece prevale in Cina, dove solo il 26% degli intervistati crede che la reputazione di una persona possa essere messa in pericolo online.

OGNI SETTIMANA NASCONO 57.000 NUOVI FALSI INDIRIZZI WEB PER COLPIRE GLI UTENTI

Ogni settimana, gli hacker creano 57.000 nuovi indirizzi Web fittizi che vengono posizionati e indicizzati sui motori di ricerca nella speranza che gli utenti, inconsapevolmente, li aprano per errore. Qualora questo dovesse accadere, il computer potrebbe essere attaccato o i dati di login inseriti su una determinata pagina potrebbero essere recapitati nelle mani dei cyber criminali.

Per realizzare tutto questo, gli hacker utilizzano una media di 375 marchi aziendali o nomi di istituzioni private di tutto il mondo, riconoscibili all'istante. eBay, Western Union e Visa sono in vetta alla lista dei più utilizzati, seguiti da Amazon, Bank of America, Paypal e il sito della finanza americana.
Circa il 65% di questi falsi siti Web sono collegati alle banche, per cercare di rubare le credenziali degli utenti quando utilizzano servizi online. Anche negozi e pagine di aste online sono molto popolari, con il 27%, ed eBay è il più utilizzato. Altre istituzioni finanziarie (come fondi di investimento o brocker di borsa) e organizzazioni governative occupano le posizioni successive, rispettivamente con il 2.3% e l'1.9%. Quest'ultima categoria è rappresentata quasi nella totalità dalla finanza americana o agenzie delle imposte. Piattaforme di pagamento (PayPal) e Internet Service Provider si stagliano al quinto e al sesto posto, mentre i siti di gaming – World of Warcraft in particolare – chiudono la classifica.

Negli scorsi anni, malware o messaggi di phishing venivano distribuiti via email, nel 2009 e soprattutto quest'anno, gli hacker hanno preferito le tecniche di Black Hat SEO, che prevedono la creazione di falsi siti web utilizzando nomi di brand famosi, etc. In questo modo, quando un utente ricerca un determinato marchio, un link al sito pericoloso apparirà tra i primi risultati. Quando si visiterà una di queste pagine, un malware potrebbe essere scaricato sul PC dell'utente, senza che questi ne sia consapevole, oppure il sito sarà molto simile all'originale e gli utenti inseriranno i propri dati, che finiranno nelle mani degli hacker.

Arriva l'MP3 DEI PAGAMENTI SICURI

Cash-Mobile della milanese 4Tech+ è un sistema di wireless payment che smaterializza totalmente la funzione di pagamento, e garantisce la massima sicurezza attraverso una tecnologia coperta da brevetto depositato. Quale sarà la prossima rivoluzione nei servizi di pagamento? Si dice la sparizione della moneta "vera" a favore di quella di plastica, ma a ben guardare il vero sconvolgimento sarà la smaterializzazione della funzione di pagamento: non più carte di plastica che si perdono e nemmeno chip di silicio che si rompono, tutto si può fare via software, e con la *massima sicurezza*. E *non nel* futuro, ma ora, ad opera di una società italiana, la milanese 4Tech+ (www.4techplus.com) specializzata in telecomunicazioni e sicurezza. Quello che l'MP3 ha fatto per la musica lo sta facendo Cash-Mobile per i pagamenti. Cash-Mobile è un servizio interfacciabile con qualunque infrastruttura di pagamento (banche, società di carte di credito, operatori innovativi come Paypal, POS) che si basa su un'architettura client-server. Il client, che risiede sul dispositivo mobile dell'utente, si occupa dell'interfaccia e della gestione della cifratura, i server delle comunicazioni con l'infrastruttura di pagamento. Il funzionamento è semplicissimo e ricorda quello tipico di una carta di credito ma... senza che alcuna carta venga strisciata o letta in altro modo (fine delle clonazioni).
Il cliente si registra sul servizio Cash-Mobile inserendo via web o in altro modo i propri dati personali, le infrastrutture di pagamento su cui vuole appoggiarsi e il suo numero di rete cellulare, e riceve un SMS contenente il link per scaricare l'applicazione client, un software scritto in Java. Il software si installa sul terminale mobile e al primo avvio si completa la procedura di registrazione: il cliente imposta un proprio PIN, che decide lui stesso. A questo punto *l'applicazione client* è abilitata.
La transazione di pagamento avviene in modo semplice, che ricorda quello tipico delle carte di credito tramite POS. Il venditore invia al servizio Cash-Mobile tramite un dispositivo mobile o un PC i dati necessari. Cash-Mobile invia sul cellulare del cliente la richiesta con gli estremi dell'acquisto, e a quel punto il cliente da' l'ok inserendo il proprio PIN segreto. La transazione viene effettuata e le relative ricevute vengono inviate sia al cliente che al venditore. Dalle prove effettuate con carico standard, il ciclo di autorizzazione dura 3-5 secondi, in quanto il cliente deve solo inserire il PIN e dare conferma, quindi non deve digitare importi, navigare tra menù, a tendina o meno, eccetera.
Tutto il sistema è basato su crittografia a doppia chiave asimmetrica e non fa circolare alcun dato sensibile. Inoltre tutte le comunicazioni tra client e server, sia in fase di registrazione/abilitazione che in fase di transazione avvengono attraverso SMS cifrati.
Rispetto ad altri servizi annunciati per il pagamento attraverso terminali mobili, Cash-Mobile offre un livello elevatissimo di sicurezza e un'assoluta semplicità d'uso, dovuta a alcuni fattori esclusivi:

- La transazione di pagamento non viene mai avviata dal cliente ma sempre dal venditore.
- La comunicazione tra client e server avviene esclusivamente tramite SMS push cifrati
- Il PIN deciso dall'utente non viene MAI condiviso in rete (nemmeno il servizio Cash-Mobile e i circuiti di pagamento lo conoscono), altro punto debole invece di molti atri sistemi
- Tutte le operazioni, una volta che il client è abilitato (procedura necessaria solo una volta), avvengono in modo trasparente per venditore e utente.

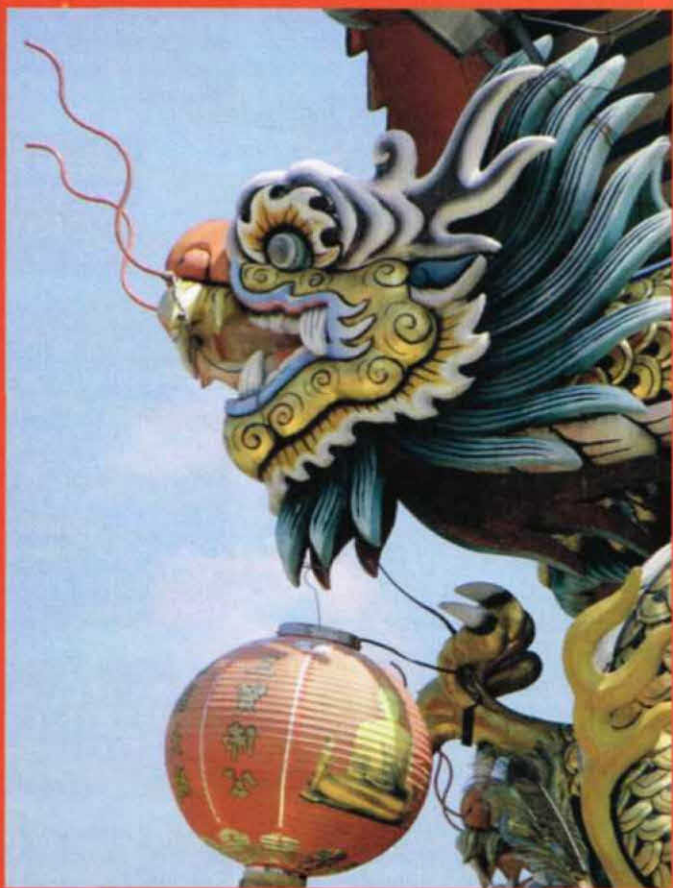
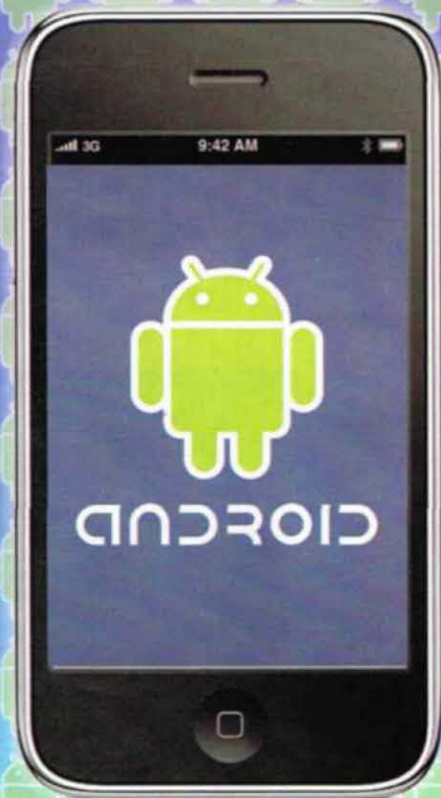
Cybercrime, ma quanto mi costi?

In Italia, l'incidenza della criminalità informatica è alta, infatti il 69% degli utenti ha subito una qualche forma di attacco online.

Nonostante la diffusione dei rischi che gli utenti devono affrontare, il tempo necessario per risolvere un problema di cybercrime è al quarto posto tra tutti i paesi studiati.

Occorrono infatti circa 36 giorni per debellare una minaccia, che sembrano un'eternità rispetto ai 9 impiegati dagli svedesi. In Europa però c'è chi fa peggio: i più lenti infatti sono i tedeschi, con una media di 58 giorni. Il cyber crimine è dispendioso anche in termini economici: eliminare il danno costa, infatti, a ciascun utente italiano circa 93 euro.

Cina sotto attacco

Gli atteggiamenti nei confronti della criminalità informatica variano da paese a paese: ad esempio in Cina, capitale mondiale del cybercrime che conta più vittime (83%), i virus e i malware sono i più temuti perché colpiscono il 65% degli utenti, al contrario in Brasile se ne sottovaluta il rischio. Anche i comportamenti volti a proteggersi dagli attacchi sono differenti: mentre gli australiani si limitano ad eliminare le mail con allegati sospetti (89%), i tedeschi preferiscono prendere precauzioni più solide, e aggiornano spesso i software di sicurezza (79%); inglesi e francesi temono soprattutto i furti di denaro e controllano spesso l'estratto conto bancario (66%), mentre gli italiani evitano di comunicare dettagli personali (67%).

di Vito Castellano
castellano.vito@gmail.com

ANDROID

In questa guida verrà spiegato come poter portare Android Froyo 2.2 su iPhone 3G jailbrekkato.
Su tratta di una guida facilmente accessibile a tutti, ma, sicuramente, chi conosce e ha fatto già pratica con Linux è un passo avanti.

PRIMA DI COMINCIARE

Per poter sfruttare questa guida occorre installare Virtual Machine o, in una partizione a parte, almeno Ubuntu.
E' consigliabile Ubuntu per i principianti perché è alla portata di tutti grazie alla sua semplicità di installazione e alla sua interfaccia con cui è rapido familiarizzare.
Unico problema da segnalare prima di iniziare il nostro tutorial è che la procedura in questione potrà richiedere, in alcuni casi, che alcuni passaggi vengano ripetuti anche più volte se tutto non va come previsto, quindi occorre non perdersi d'animo e avere una certa determinazione per il conseguimento del risultato finale.

A PROPRIO RISCHIO E PERICOLO

Come tutte le procedure non autorizzate dalla casa madre, anche questa determina un decadimento della garanzia dell' iPhone nonché una piccole dose di rischi correlati, anche se si tratta di un procedimento piuttosto diffuso e sperimentato. Comunque è sempre bene essere consapevoli dei rischi intrinseci. Come si dice: uomo avvisato...

AVRÒ ANCORA IPHONE OS SUL MIO IPHONE?

Certo grazie ad OpenIboot avremo la possibilità di avere un vero e proprio DualBoot.
Attenzione per far sì che il procedimento funzioni bisogna avere un iPhone jailbrekkato con firmware non superiore a 4.0 . Se avete già eseguito degli aggiornamenti, ad esempio 4.0.1 o 4.0.2,
bisogna effettuare il downgrade. Se proprio lo dovete fare è consigliabile tornare a 3.1.3 . Vediamo come eseguire il downgrade.

Per prima cosa occorre procurarsi i seguenti strumenti :

- iPhone1,2_3.1.3_7D11_Restore.ipsw (se volete passare ad OS 3.1.3)
- iRecovery (Per Windows)

Adesso è possibile collegare l'iPhone al computer e metterlo in modalità Recovery. Per mettere l'iPhone in modalità Recovery basta semplicemente collegarlo al proprio computer e quindi schiacciare contemporaneamente il tasto "Home" ovvero il tasto centrale dell'iPhone e il tasto "Spegni/Accendi" ovvero il tasto

che normalmente si schiaccia per accendere il telefono. Dopo circa 3-4 secondi dovrebbe spegnersi. Tenete ancora premuti i due tasti finché non visualizzate il logo della mela, a quel punto non dovrete più schiacciare il tasto "Power" ma tenete ancora premuto il tasto "Home" finché non visualizzerete il logo di iTunes che vi indica di attaccare il telefono al computer. Alla fine dovreste trovarvi nella situazione come riportato nella figura

Adesso il vostro iPhone si trova in modalità " Recovery ", quindi apriamo iTunes.
Esso vi avvertirà che ha rilevato il telefono in modalità ripristino, e ovviamente vi chiederà di ripristinarlo.Tenendo premuto il tasto " Shift " della tastiera e facendo click con il mouse sul bottone "Ripristina " avrete la possibilità di selezionare il vostro OS.
Nel nostro esempio andremo a selezionare iPhone1,2_3.1.3_7D11_Restore.ipsw.
Inizierà così il ripristino, ad un certo punto iTunes riscontrerà uno dei seguenti errori (1011, 1013, 1015) occorrerà quindi eseguire queste semplici operazioni:
Prima di tutto installiamo la libreria libusb (libusb-win32-filter-bin-0.1.12.2.exe), la troverete nella cartella di iRecovery.
Attenzione per gli utenti di Windows Vista/7: Prima di installare la libreria USB, tasto destro sul file libusb-win32-filter-bin-0.1.12.2.exe, impostiamo la compatibilità con Windows XP SP2 e spuntiamo la voce "esegui come amministratore", poi diamo ok.
Adesso andremo ad avviare il Prompt dei comandi.
Per avviare il prompt dei comandi:

Start>esegui >digitare "cmd" ovviamente senza virgolette (Windows XP).
Start >cercare esegui>una volta trovato >digitare "cmd" ovviamente senza virgolette (Windows Vista/7).

-Adesso bisognerà posizionarsi nella cartella che contiene iRecovery questo dipenderà da dove avrete posizionata la cartella, il consiglio è di tenerla a portata di mano sul Desktop e digitare nel Prompt dei comandi

cd Desktop\"nome cartella che contiene il programma iRecovery "

ad esempio

```
cd Desktop\iRecovery <
premere invio >
```

adesso che ci siamo posizionati nella cartella è arrivata l'ora di eseguire iRecovery dando questo comando:

```
iRecovery.exe -s < premere
invio >
```

e di conseguenza diamo questi comandi

```
setenv auto-boot true <
premere invio>
saveenv <premere invio>
/exit <premere invio>
```

In questo modo l'iPhone uscirà dalla modalità "Recovery" e lo vedrete riavviarsi. Adesso avete in mano il vostro iPhone con Firmware 3.1.3
E' arrivato il momento di effettuare il JailBrekkare del "melafonino", si consiglia di utilizzare RedSnow anche perché se volete mettere Android, effettuando il jailbreak con altri programmi (Spirt, BlackRa1n ecc.) andrà in conflitto e non riuscirete ad installarlo.
Non staremo qui a spiegarvi come effettuare il Jailbreak con RedSnow esistono numerose guide in rete comunque se proprio non dovreste riuscirci potete contattare la redazione.
Bene adesso abbiamo il nostro iPhone 3G OS 3.1.3 Jailbrekkato con Cydia installato , apriamo Cydia e andiamo a ricercare prima di tutto OpenSSH e installiamolo questo programma ci permetterà di aprire una sorta di comunicazione che ci consentirà di trasferire i nostri file dal computer all'iPhone.
Andremo ad installare anche "Sbsetting" questo programma ci aiuterà a mantenere sotto controllo lo stato attivo di OpenSSH, per utilizzare questo programma basterà semplicemente far scivolare il dito in senso orizzontale sulla barra di stato.

Ora abbiamo tutto il necessario checi serve per poter installare Android su iPhone.
Passiamo ad ubuntu e procuriamoci i seguenti file:

- Openiboot
- idroid
- Estrazione

Tutti questi file li troverete nel blog all'indirizzo http://idroidshare.blogspot.com/. Adesso per prima cosa scompattiamo tutti gli archivi (idroid-1.0.2.tar.gz , openiboot.tar.gz , *estrazione.tar.gz)* e portiamo le cartelle sulla Scrivania. Ricordatevi bene che su Ubuntu il desktop si chiama Scrivania Adesso accendete il Wi-fi su iPhone e anche il servizio SSH potrete attivare questi servizi tramite "Sbsetting" e annotatevi il vostro indirizzo IP. Ora su Ubuntu andremo in

`Risorse>Rete`

Qui troverete il vostro iPhone, facendo click su di esso apparirà una finestra dove inseriremo

Nome utente : root
Password : alpine (è la password di default di OpenSSH)

Bene adesso siamo dentro seguiamo questo percorso spostandoci tra le cartelle

`Private>Var`

Prima di tutto creiamo una cartella e la rinomineremo "sdcard" questa cartella andrà ad emulare appunto la sd card, lo spazio dove potremmo inserire i nostri file.
Sempre in "Private ? Var" sposteremo tutta la cartella "idroid" ovvero la cartella che contiene : android.img.gz , cache.img , system.img , userdata.img , zImage .
Dopo aver caricato questi file (le immagini di Android), andiamo a caricare i driver necessari per il funzionamento del Wi-Fi e del Touch.
Per ricavare i driver useremo un metodo di estrazione chiamato "ninn's extraction technique" questo "programma" lo troveremo nella cartella estrazione con il nome di extractiontechnique0.2.sh .

COME AVVIARE IL PROGRAMMA?

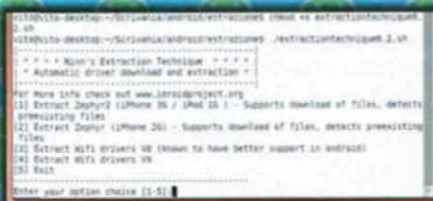
Innanzi tutto avviamo il terminale lo troveremo in

Applicazioni
`>Accessori>Terminale`

aperto il terminale daremo questi comandi

cd Scrivania (ci spostiamo sul Desktop)
cd estrazione (ci spostiamo nella cartella estrazione)

```
chmod +x
extractiontechnique0.2.sh
./
extractiontechnique0.2.sh
```

e ci ritroveremo in una condizione come nell'immagine in figura.

A questo punto noi andremo ad estrarre " Zephyr2" ovvero il punto 1 e il punto 3, i driver "Wi-Fi".
Non c'è molto da spiegare, ovviamente basta inserire su terminale prima " 1 " e dopo che avrà estratto il primo driver andremo ad inserire "3" .
Adesso nella cartella estrazione ci troveremo 3 nuovi file ovvero :

`sd8686.bin, sd8686_helper.bin, zephyr2.bin.`

Fatto questo ritorniamo sulla nostra cartella dell'iphone :

`Private >Var`

qui creeremo una nuova cartella chiamandola "firmware" in modo che il risultato sia così

`Private >Var >firmware`

Che copieremo anche dentro la cartella "idroid". Alla fine il risultato deve essere questo

```
Private> Var >firmware
Private>Var>idroid>firmware
```

dove in questa cartella firmware inseriremo i nostri 3 driver che abbiamo estratto poco fa (sd8686.bin , sd8686_helper.bin , zephyr2.bin) .

Bene siamo giunti agli ultimi passaggi adesso ricordate all'inizio della guida quando vi avevamo fatto mettere in modalità Recovery l'iPhone? Ecco, ora dovrete di nuovo eseguire quella procedura.
Una volta messo in modalità Recovery apriamo il terminale di Ubuntu e digitiamo
cd Scrivania (con questo comando ci sposteremo sul Desktop)

```
sudo apt-get install
libusb-0.1-4 (ci chiederà
di confermare e noi
daremo una bella "Y" per
confermare)
cd openiboot ( ci
spostiamo nella cartella
openiboot contenete 3
file : loadibec , oibc ,
Openiboot.img3 )
sudo ./loadibec openiboot.
img3
```

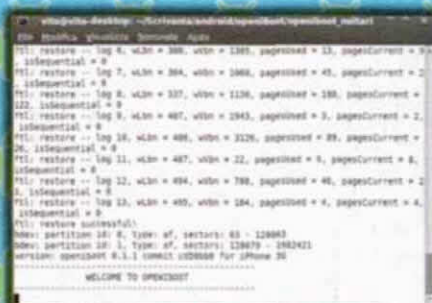
Dando questo comando troveremo evidenziata sullo schermo del nostro iPhone una schermata come quella proposta in figura, che ci mostrerà tre possibili selezioni: iPhone OS , Console e iDroid.

Adesso andremo a spostarci su console attraverso i tasti dell'iPhone per regolare il volume, quindi spostiamoci su console e schiacciamo una sola volta il tasto " Home " del nostro iPhone. Inizieranno ad apparire sullo schermo del nostro iPhone tante stringhe di codice, non stiamo qui a spiegarvi cosa rappresentano, a voi interessa solo che l'ultima stringa stampi

```
" WELCOME TO OPENIBOOT "
```

adesso sempre da terminale andremo a digitare

```
5. sudo ./oibc
```

ci ritroveremo in una situazione dove le stesse stringhe stampate nel terminale appariranno contemporanemante sull'iPhone come in figura

Se siete arrivati allo stesso risultato (ottimo) digitiamo l'ultimo comando:

```
install ( e premiamo invio )
```

ci stamperà una serie di stringhe come questa

```
install
Backing up your NOR to
current directory as
norbackup.dump
Fetching NOR backup.
file sent.
NOR backed up, starting
installation
Installing Images...
Reading images...
Reading: ibot (286912
bytes)
Reading: ibox (171328
bytes)
Reading: dtre (43968 bytes)
Reading: logo (10624 bytes)
Reading: recm (48896 bytes)
Reading: nsrv (21504 bytes)
Reading: bat0 (57792 bytes)
Reading: bat1 (66368 bytes)
Reading: glyC (21376 bytes)
Reading: glyP (20352 bytes)
Reading: chg0 (20736 bytes)
Reading: chg1 (25920 bytes)
Reading: batF (77120 bytes)
Performing upgrade...
(283512 bytes)
Total size to be written
872896
Flashing...
Flashing: ibot (a1f61d8,
286912 bytes)
Flashing: ibox (a08d6c0,
171328 bytes)
Flashing: dtre (a193748,
43968 bytes)
Flashing: logo (a0879a8,
10624 bytes)
Flashing: recm (a19e310,
48896 bytes)
Flashing: nsrv (a1aa218,
21504 bytes)
Flashing: bat0 (a1af620,
57792 bytes)
Flashing: bat1 (a1bd7e8,
66368 bytes)
Flashing: glyC (a1cdb30,
21376 bytes)
Flashing: glyP (a1d2eb8,
20352 bytes)
Flashing: chg0 (a1d7e40,
20736 bytes)
Flashing: chg1 (a1dcf48,
25920 bytes)
Flashing: batF (a1e3490,
77120 bytes)
Free space after flashing
28224
Done with installation!
Refreshed image list
Images installed
Setting openiboot
version...
Successfully loaded bank1
nvram
Successfully loaded bank2
nvram
Openiboot installation
complete.
```

Ora possiamo digitare

```
reboot ( riavvierà
l'iPhone )
```

LANCIAMO IDROID

Qui si conclude l'installazione. Adesso avremo la possibilità di avviare "iDroid" spostiamoci su di esso sempre con i tasti della regolazione del volume , dopodichè, premendo sul tasto " Home " una sola volta, vedremo caricarsi tutto il " Kernel " e successivamente " Android " fino ad arrivare a questo risultato

Qui finisce la mia guida per qualsiasi problema o informazione scrivete alla redazione o passate direttamente al blog dell'autore

```
blog : http://idroidshare.
blogspot.com/
```

Se lascerete qualche commento vi verrà risposto in tempi brevi.

di Spirit
redazione@hackerjournal.it

VIRUS: LA SINTOMATOLOGIA

INFECTED

COME CAPIRE SE UN COMPUTER È AFFETTO DA UN VIRUS O PEGGIO? QUALCHE PICCOLO SINTOMO PUÒ ESSERE INDICATIVO.

I virus dell'influenza è facilmente identificabile: la temperatura sale, si avverte uno stato di generale malessere accompagnato anche da altri fenomeni legati alla patologia e non ci si regge in piedi.

Questo per gli umani. Tuttavia capire se un computer è vittima di un virus diventa assai più complesso. Certo ci sono gli antivirus che in qualche modo ci tutelano, un po' come il vaccino per l'influenza, però il rischio di contrarre qualche insidioso parassita informatico è tutt'altro che scongiurato.

Oggi gli autori di virus, worm, Trojan e spyware si sono spinti fino a riuscire a cancellare i propri codici e nascondere ciò che i propri programmi stanno eseguendo su di un computer infetto. Vediamo quindi di capire, in questo breve articolo, quali sintomi devono fare scattare l'allarme e come ci si deve comportare.

PREVENIRE È MEGLIO CHE CURARE

La prima regola è quella di prevenire, sempre e comunque. Quindi installare un sistema di protezione che si aggiorni periodicamente, installare le patch e gli aggiornamenti del sistema operativo e delle applicazioni appena questi vengono rilasciati e tenere sempre una copia di backup dei dati (non si sa mai). Detto questo non si può avere la matematica certezza di essere comunque al sicuro da qualsiasi rischio, il computer può venire compromesso comunque e, in questo caso, dà qualche segnale che deve essere correttamente interpretato perché si tratta spesso di sintomi che possono essere generati anche da problemi di hardware e/o software.

I SINTOMI TIPICI

Ecco un elenco dei sintomi che devono in qualche modo far sorgere il dubbio sullo stato di "salute" del computer:

Il computer si comporta in maniera strana e inusuale.
Compaiono immagini o messaggi inaspettati.
Emette dei suoni strani, a caso.
I programmi si aprono da soli.
Il firewall comunica che un'applicazione ha cercato di connettersi a Internet (e l'applicazione non è un programma che utilizzate).

Gli amici dicono che hanno ricevuto dei messaggi e-mail dal vostro indirizzo di posta elettronica, che non sono mai stati nè scritti, nè inviati.
Il computer si blocca spesso o i programmi impiegano molto tempo ad aprirsi.
Si ricevono molti messaggi di "errore di sitema".
Il sistema operativo non carica all'avvio del computer.
Alcuni file o cartelle sono stati modificati.
Viene segnalato un accesso all'hard disk (tramite il lampeggiare del display) quando non risulta che ci sia alcun programma aperto.
Il browser si comporta in maniera imprevedibile, ad esempio, non si riesce a chiudere una finestra.

COSA FARE?

Se riscontrate qualcuno dei sintomi sopra descritti, non fatevi prendere dal panico. Potreste semplicemente avere un problema dell'hardware o a livello di software, e non un virus, un worm o un Trojan. Ecco, in ogni caso, cosa fare:

SCONNETTETE IL VOSTRO COMPUTER DA INTERNET

Se il sistema operativo non carica, avviate il computer in Modalità di Sicurezza (quando accendete il computer, premete per qualche secondo il tasto F8, e poi selezionate la Modalità di Sicurezza dal menu che apparirà), oppure eseguite l'avvio da un CD di soccorso.
Assicuratevi che le firme anti-virus siano aggiornate. Se possibile, non scaricate gli aggiornamenti usando il computer che temete sia infetto, ma un altro (ad esempio quello di un amico). Ciò è importante: se il vostro computer è infetto e vi connettete tramite esso ad Internet, un programma maligno potrebbe inviare informazioni importanti e confidenziali ad un hacker remoto, o propagarsi auto-inviandosi a quegli indirizzi e-mail che sono archiviati sul vostro computer.
Se rimuovere il malware dovesse risultare problematico, controllate il sito web del vostro anti-virus per avere informazioni o accesso ad utility dedicate che potrebbero essere necessarie per rimuovere singoli programmi maligni. Se il computer è connesso ad una rete locale, disconnettetelo da tale rete.

ESEGUITE UNA SCANSIONE DI TUTTO IL COMPUTER

Se dovesse venire rilevato un programma nocivo, seguite i suggerimenti forniti dal vostro vendor del software di Internet security. Un buon programma di sicurezza prevede l'opzione di disinfezione degli oggetti compromessi, la possibilità di metterli in quarantena e di eliminare worm e Trojan. Tali soluzioni sono in grado di creare un file di notifica che elenca i nomi dei file infetti e dei programmi malware trovati sul computer.
Se il vostro software di protezione non trova nulla, allora il vostro computer è quasi sicuramente sano. Controllate l'hardware e i software installati sul computer (è bene rimuovere i software privi di licenza e qualsiasi file spazzatura) e assicurarsi di avere installata l'ultima versione di sistema operativo e le patch di sicurezza delle applicazioni.
Se necessario, contattate l'assistenza tecnica del produttore del vostro software di protezione anti-virus per ulteriori indicazioni. Potete anche chiedere di sottoporre ad analisi un file campione.

di Massimiliano Rinaldi
redazione@hackerjournal.it

"INCENERIRE" LO SPAM CON NAPALM

ANTISPAM

Napalm è un efficace anti-spam di minuscole dimensioni scritto in TLC.

In questo articolo presentiamo una semplice ma efficace contromisura allo spamming, sviluppata sotto forma di script in Tcl.

Ci occuperemo prima di spamming in generale, poi dei dettagli della soluzione proposta; l'articolo dovrebbe quindi risultare interessante sia a quanti si trovino ad avere a che fare con lo spamming, sia a quanti desiderino fare un esercizio di programmazione con il linguaggio Tcl.

Le caratterische della soluzione che qui proponiamo sono:

- **le ridottissime dimensioni (meno di 120 righe di codice, facilmente modificabili);**
- **l'indipendenza dalla piattaforma (lo script funziona indifferentemente sotto Linux e Windows senza modifiche);**
- **l'efficienza (i messaggi vengono eliminati sul server senza spreco di banda);**
- **la flessibilità (le regole sono contenute in un semplice file di testo che si può modificare liberamente e condividere con amici e colleghi);**
- **la sicurezza (i messaggi cancellati vengono registrati in un log).**

DI SPAM IN SPAM

Per spam o spamming si intende la posta elettronica non richiesta e indesiderata, solitamente a carattere pubblicitario. Si tratta di un fenomeno abbastanza recente, che negli ultimi anni mostra una forte crescita. Alcuni utenti, per lo più quanti ne sono stati toccati solo marginalmente, non si sono ancora fatti un'opinione a riguardo

del fenomeno (o peggio ancora lo sottovalutano), mentre altri (quelli più smaliziati) sono sensibili al problema, e sono restii a divulgare il proprio indirizzo di posta elettronica.
Sappiamo infatti che ogni volta che ci registriamo con il nostro indirizzo e-mail su un sito, c'è una certa probabilità che esso abusi del nostro indirizzo, non necessariamente vendendolo a spammer, magari anche soltanto per inviarci la newsletter settimanale del sito stesso, che però ha carattere pubblicitario e può addirittura non prevedere un modo per disdire l'iscrizione. Ci sono poi comportamenti molto meno tollerabili, come quelli di chi, con software appositamente preparato, scandaglia newsgroup, forum di discussione e altri servizi alla ricerca di indirizzi validi, rastrellando tutti quelli che vengono trovati, al fine di usarli in proprio o venderli ad acquirenti senza scrupoli. E' proprio per eludere questi software che molti messaggi sui newsgroup indicano l'indirizzo del mittente in maniera camuffata, per esempio come nome(at)dominio(dot)it.
Combattere lo spamming è possibile con contromisure legali o tecniche; queste ultime si risolvono nell'utilizzare account e-mail filtrati, con filtri sul lato server o con contromisure sul lato client. Esaminiamo queste soluzioni nel dettaglio.
Per quanto riguarda le azioni legali, in caso di abuso del proprio indirizzo di posta, in Italia è possibile ricorrere all' Autorità Garante per la protezione dei dati personali. Sulla rete si trovano alcune esperienze a riguardo, tutte concordi nell'affermare la tempestività e l'efficacia degli interventi del Garante, che può anche comminare risarcimenti pecuniari ai responsabili dello spamming, da versare alle "vittime". Questo approccio è però impraticabile sui "grandi volumi" (richiede infatti almeno l'invio di un paio di raccomandate per ciascun caso) ed è scarsamente efficace nei confronti di spammer situati all'estero.
La prima tipologia di contromisure tecniche è l'uso di servizi che offrono un account di posta elettronica filtrato, alcuni gratuiti (come despammed.com) e altri a pagamento, con eventuali servizi in più a valore aggiunto. Invece, nel caso in cui siate voi stessi a gestire "in proprio" il mail exchanger del vostro dominio, esistono pacchetti molto interessanti (come SpamAssassin) che lavorano in stretta cooperazione con il mail transport agent e agiscono contro lo spam in maniera molto efficace, proprio perché centralizzati. Infine vi sono le contromisure su client, che tutti possono applicare: ormai gran parte dei programmi client di posta elettronica (come Outlook Express, Eudora o KMail) sono dotati di una sezione in cui è possibile definire regole per il trattamento della posta; fra queste regole, ci sono anche quelle distruttive contro lo spam.

LA MIA ESPERIENZA PERSONALE

Al momento ricevo quotidianamente circa un centinaio di messaggi di spamming. Il mio log del Napalm contiene in totale diciottomila messaggi eliminati, distribuiti in maniera abbastanza uniforme. Ricevo ogni giorno proposte di carte di credito, metodi per far soldi velocemente, vacanze, lotterie, materassi, assicurazioni, cure dimagranti, biglietti aerei, cartucce di inchiostro, siti con materiale erotico, telefoni cellulari, protesi mammarie, decoder per tv digitale, kit per allergia nasale, DVD, viagra, titoli di studio (improbabili), immobili, lettori MP3, servizi araldici, mobilio, campioni di caffè, software, scooter elettrici, oroscopi, macchine fotografiche, automobili, sigari, servizi legali, prodotti farmaceutici, carta igienica parlante (giuro!), prestiti, gioielli, tessere sconto, fiori, cosmetici, giocattoli, cioccolato e mille altre cose che non sto ad elencarvi per non abusare della vostra pazienza.
Prima di realizzare Napalm ho provato svariate soluzioni, fra cui i filtri di Outlook ed Eudora, e il servizio despammed.com, ma nessuna mi ha soddisfatto completamente. Nei primi due casi (l'uso di filtri all'interno del client di posta), mi infastidiva il fatto che l'applicazione delle regole venisse fatta sui messaggi una volta che questi erano ormai già stati scaricati sul client. In tal modo è pur vero che tali messaggi vengono spostati nel cestino, e quindi non disturbano l'utente, tuttavia vengono comunque scaricati dal server, e il danno è fatto (dispendio di banda e spazio su disco; aumento dei tempi e dei costi di connessione in caso di collegamento telefonico).
Preferisco una soluzione che sia in grado di individuare i messaggi indesiderati sul server, e possa eliminarli direttamente lì, senza perdere tempo nello scaricarli o sprecare spazio su disco. In entrambi i programmi citati parrebbe che le opzioni permettano di eliminare i messaggi dal server, ma in realtà ciò non avviene.
Nell'ultimo caso (l'uso di un servizio automatico di filtraggio) la mia esperienza concerne il servizio despammed.com. Si tratta di un servizio gratuito a cui ciascuno si può iscrivere, ottenendo un indirizzo di posta elettronica del tipo proprionome@despammed.com, il quale è soggetto a filtraggio. Si può inoltre impostare tale casella di posta in modo da fare forwarding verso un altro indirizzo, oppure si può

consultarne il contenuto via web. A mio parere il servizio non è soddisfacente, in quanto le regole imposte dai gestori sono così restrittive che non solo eliminano tutti i messaggi di spamming, ma anche gran parte di quelli desiderati. Non vi è inoltre modo di esaminare o modificare tali regole, né di ottenere un tracciato con l'elenco dei messaggi eliminati.
C'è un ulteriore motivo di insoddisfazione nell'uso delle regole di filtraggio con i client di posta che ho provato: impostare una regola è una operazione inutilmente laboriosa. Alla ricezione di dieci nuovi messaggi non catturati dalle regole correnti, impostare ulteriori dieci regole richiederebbe gran lavoro di mouse e tastiera senza motivo. Spesso non è possibile esportare le regole, né tanto meno farsi spedire da un amico le sue e aggiungerle alle proprie.
Napalm, pur nella sua minimalità, è fatto in modo da soddisfare tutte le richieste di cui ho parlato nei paragrafi precedenti: agisce identificando e rimuovendo i messaggi indesiderati dal server senza scaricarli; registra con precisione mittente, destinatario e oggetto di tutti i messaggi eliminati; rende il compito di aggiunta, modifica e rimozione delle regole semplice come aggiungere, modificare o togliere una riga da un semplice file di testo; permette agevolmente di inviare le proprie regole ad amici e colleghi e ricevere le loro.

INSTALLAZIONE DI NAPALM SOTTO LINUX

Una volta scaricato il file Napalm-0.3.tar.gz (dall'indirizzo:http://www.scarpaz.com/Napalm/Napalm-0.3.tar.gz) nella vostra directory home, scompattatelo digitando il comando:

```
tar xvzf Napalm-0.3.tar.gz
```

Tale comando crea una sottodirectory della vostra home che si chiamerà Napalm. Tenete presente che per il corretto funzionamento di Napalm sotto Linux è obbligatorio che l'installazione avvenga proprio in ~/Napalm. Questo vincolo è pensato per rendere possibile l'invocazione periodica automatica di Napalm tramite cron, funzionalità che verrà descritta meglio nel seguito. (Tale vincolo non si applica sotto Windows, dove non esiste cron).
Cercate, nella directory citata, il file Napalm.conf. Dovrete modificarne il contenuto in modo da riflettere le vostre impostazioni di posta elettronica.
Il formato del file quello che segue:

```
set hostname "popmail.
provider.it"
set port 110
set username "il_vostro_
nome_utente"
set password "la_vostra_
password"
set verbose 0
```

Le vostre modifiche faranno in modo da sostituire alle stringhe "popmail.provider.it", "il_vostro_nome_utente" e "la_vostra_password" rispettivamente il nome del server POP3 del vostro provider, il nome utente a voi attribuito e la relativa password. Potete anche modificare la porta se non è quella standard (la 110). A questo punto potete collaudare le impostazioni di Napalm semplicemente digitando:

```
./Napalm.tcl
```

(Sotto shell "esotiche" potrebbe essere necessario digitare tclsh Napalm.tcl).

Attenzione: il file Napalm.conf contiene la vostra password di accesso in formato leggibile; se la vostra macchina ospita più utenti, per la vostra sicurezza, fate in modo che Napalm.conf non sia leggibile dagli altri, per esempio con il comando:

```
chmod 600 Napalm.conf
```

Si può fare in modo che Napalm venga eseguito periodicamente senza necessità di intervento da parte vostra. Se le vostre impostazioni sono correttamente collaudate, potete chiedere a cron di lanciare Napalm ogni (ad esempio) quarto d'ora. Per fare ciò assicuratevi che il demone cron sia installato sulla vostra macchina, verifica che con RedHat e simili si compie con il comando:

```
service crond status
```

Se cron non è in esecuzione, è necessario procedere alla sua configurazione e al suo avvio (numerose FAQ e HowTo descrivono come).
Preparate quindi nella vostra home un file chiamato, per esempio, crontab, che contiene le linee seguenti:

```
0 0-23/1 * * * /root/
Napalm/Napalm.tcl
15 0-23/1 * * * /root/
Napalm/Napalm.tcl
30 0-23/1 * * * /root/
Napalm/Napalm.tcl
45 0-23/1 * * * /root/
Napalm/Napalm.tcl
```

oppure

```
0 0-23/1 * * * /home/nome_
utente/Napalm/Napalm.tcl
15 0-23/1 * * * /home/
nome_utente/Napalm/Napalm.
tcl
30 0-23/1 * * * /home/
nome_utente/Napalm/Napalm.
tcl
45 0-23/1 * * * /home/
nome_utente/Napalm/Napalm.
tcl
```

a seconda che siate root o che siate l'utente nome_utente. Il significato delle quattro istruzioni qui sopra citate è il seguente: a ogni minuto zero, quindicesimo, trentesimo, e quarantacinquesimo di ogni ora (dalle zero alle ventitrè) di ogni giorno, mese e anno, venga eseguito Napalm. Se volete approfondire la sintassi del file, è sufficiente consultare la manpage di crontab.
A questo punto basta indicare a cron di acquisire le nuove istruzioni con il comando:

```
crontab crontab
```

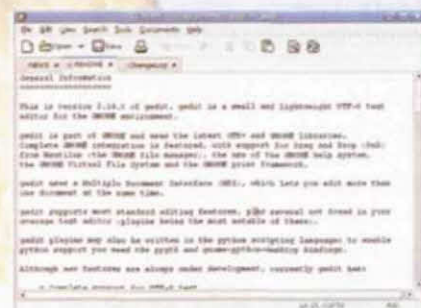
Per l'uso di Napalm con cron si consiglia di cambiare in 1 lo 0 che si trova impostato nell'ultima riga del file Napalm.conf, associato alla variabile verbose, in tal modo verranno registrati in Napalm.conf non solo i messaggi cancellati, ma anche tutti gli eventi di partenza periodica, messaggi non cancellati ed eventuali condizioni di errore.

INSTALLAZIONE DI NAPALM SOTTO WINDOWS

Ricordo che Napalm è uno script in Tcl: per eseguirlo serve quindi che un interprete Tcl sia installato nel sistema. Normalmente tutte le distribuzioni Linux in circolazione lo comprendono (per verificarlo è sufficiente aprire una shell e digitare tclsh), mentre per i sistemi Microsoft occorre procurarsene uno, per esempio ActiveTcl, disponibile gratuitamente all' indirizzo: http://aspn.activestate.com/ASPN/Downloads/ActiveTcl/. Una volta installato l'interprete, diciamo in C:\Programmi\Tcl, installare Napalm si riduce a scompattare il file Napalm-0.3.tar.gz in una directory a vostra scelta (per esempio C:\Programmi\Napalm), con un qualsiasi programma per la manipolazione di archivi (va benissimo anche WinZip), quindi configurare il file Napalm.conf esattamente come indicato nel paragrafo precedente (mediante un comune editor di testi, per esempio il notepad fornito con Windows) e creare un collegamento come segue:

```
1.tasto destro sul desktop
> crea collegamento
2.nel campo percorso
immettiamo C:\Programmi\
Tcl\bin\tclsh80.exe (fig. 1)
3.nel campo nome inserite
Napalm o altro nome a
piacere (fig. 2)
4.ricliccate con il tasto
destro sul link Napalm e
selezionate Proprietà dal
menu contestuale
5.aggiungete al campo
destinazione "C:\Programmi\
Napalm\Napalm.tcl", e
impostate la directory
corrente (il campo da) come
indicato, cioè pari a "C:\
Programmi\Napalm" (fig. 3)
6.impostate i font e la
larghezza della finestra
in modo che sia possibile
visualizzare un buon numero
di linee e colonne (fig. 4)
7.Il risultato è in figura
5.
```

CONFIGURAZIONE DELLE REGOLE

Nella directory dove avete installato Napalm, troverete anche un file Napalm.rules: questo file contiene le regole che definiscono quali messaggi debbano essere cancellati. E' vostra responsabilità modificare queste regole coscienziosamente, tenendo presente che regole sbagliate possono causare l'eliminazione indesiderata di messaggi "buoni". La modifica può avvenire con un qualsiasi editor di testo disponibile sotto Linux o Windows, come gedit (in figura), emacs, vi, oppure con notepad sotto windows. Le regole vanno definite nel formato che segue:

```
kill $from "*dealsonline*"
kill $from "*Offers*"
kill $from
"*tremendousrewards*"
kill $from "*sweepsclub*"
kill $subject
"*Lose*Pounds*"
kill $subject "*Open Any
Lock with*"
```

In particolare ciascuna linea del file rispetterà la sintassi che segue:

```
kill $nome_campo "pattern"
```

dove *nome_campo* indica il campo dell'intestazione del messaggio da controllare (può essere uno dei seguenti: from, to, subject); e pattern è una stringa che viene confrontata con il contenuto di tale campo (senza riguardo per le minuscole/maiuscole). Se il confronto ha esito positivo, il messaggio viene eliminato. I pattern possono contenere i simboli speciali punto interrogativo '?' e asterisco '*', i quali riconoscono rispettivamente ogni carattere e ogni stringa di caratteri, anche la stringa vuota. Ad esempio, il pattern "pi?na" riconosce le stringhe "piena", "piana", "pinna", "pigna", etc.; mentre il pattern "pi*na" riconosce tutte le stringhe precedenti ma anche "pina", "piccolina" e addirittura "pisa non è una città

campana". Per maggiori dettagli sul pattern matching effettuato da Napalm, consultate la guida del comando string del linguaggio Tcl; sotto Linux è facilissimo, basta digitare man n string; sotto Windows invece basta lanciare

```
Start > Programmi > Tcl >
Tcl help
```

e cercare il comando string.

Le informazioni interessanti sono al paragrafo "string match". Impostare le regole giuste per voi è, fra tutte quelle descritte, l'operazione meno banale e più creativa.

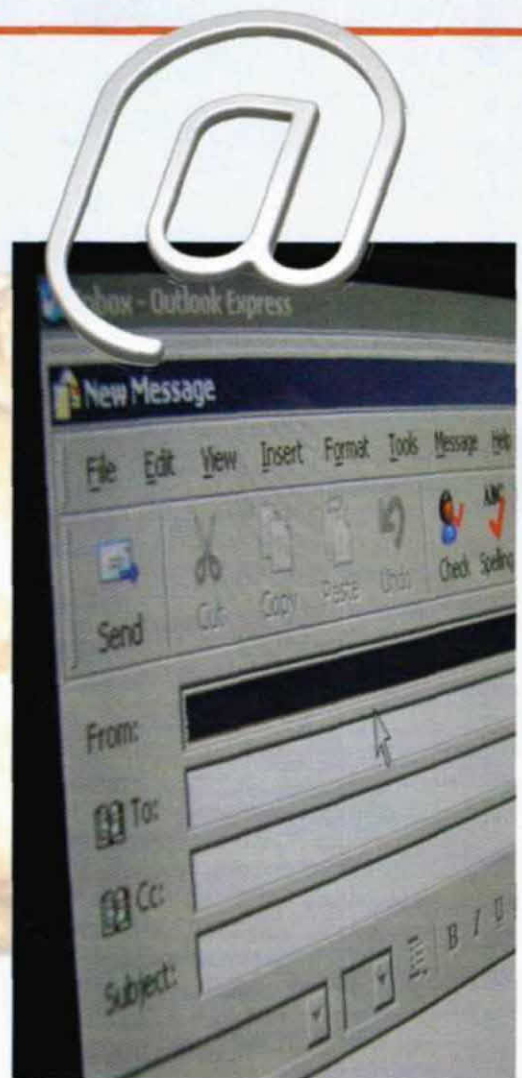
E' necessario specificare regole sufficientemente generali affinché un numero limitato di esse catturino tanto spam, facendo però attenzione a fare in modo che tali regole non corrispondano anche a messaggi legittimi. Ogni giorno qualche messaggio indesiderato riuscirà a infiltrarsi fra le regole correnti, e dovrete aggiungerne a mano qualcuna, ma una volta giunti "a regime", la stragrande maggioranza dello spam verrà bloccato.

COME USARE NAPALM

L'uso di Napalm è banale: sotto Windows basta un doppio-click sull'icona, sotto Linux si può lanciare da terminale o con un collegamento sul desktop, da creare secondo le procedure consuete che dipendono da Gnome o KDE. In entrambi i casi, se la configurazione è stata specificata correttamente, vedrete una serie di messaggi come quelli in figura:

```
/ [2009-10-20 20:20:56]
Napalm entering war
theater, Sir.
! Connecting to host:
popmail.libero.it, port:
110
| Server: [+OK POP3 PROXY
server ready (6.5.001)<2B
49EC1753641E6A0E69F6A2C614
864A3EBF7E6D@pop.provider.
com>]
| Server: [+OK Password
required]
| Server: [+OK 11
messages]
| 01: From: ["sender1"
<sender1@mail.com>] To:
[you <name@provider.com>]
Subject: [subject1]
| 02: From: ["sender2"
<sender2@mail.com>] To:
[you <name@provider.com>]
Subject: [subject2]
...
| 08: From: ["Bad Guys"
<badguys@mail.com>] To:
[you <name@provider.com>]
Subject: [spam]
X Napalm killed 08: From:
["Bad Guys" <badguys@
mail.com>] To: [you <name@
provider.com>] Subject:
[spam]
| 09: From: ["sender9"
<sender9@mail.com>] To:
[you <name@provider.com>]
Subject: [info]
| Server: [+OK POP3 server
closing connection]
\ [2009-10-20 20:21:00]
Napalm completed
operations with 1 confirmed
killings, Sir.
```

Ogni volta che un messaggio viene eliminato, Napalm stampa una linea di messaggio che inizia con una X, e aggiunge le informazioni riguardanti il messaggio eliminato al file Napalm.log, in un formato simile al seguente:

```
[2009-10-21 14:50:05] X
From: ["spammer" <spammer@
mail.com>] To: [me@
provider.com] Subject: [5
Books For $1 Plus a FREE
Attach] Size: [4573 bytes]
```

In questo modo rimarrà una traccia di tutti i messaggi eliminati, e in caso abbiate commesso un errore nelle regole sarà possibile almeno avvisare il mittente e chiedergli gentilmente di inviare ancora il messaggio.

CONCLUSIONI

Oltre ad essere un utile mini-strumento, Napalm è un interessante esempio di come, grazie al linguaggio Tcl, si possa realizzare efficacemente un client POP3 in poche righe di codice. Lo spazio a disposizione non ci consente di trattare i dettagli implementativi; li rimandiamo ai lettori interessati, che potranno anche modificare lo strumento in modo da adattarlo alle proprie esigenze.

Andrea Draghetti
info@oversecurity.net

ANALISI DEL DLL LOAD HIJACKING

Questo articolo ha l'intenzione di spiegare la vulnerabilità del DLL Load Hijacking scoperta recentemente dalla società di sicurezza Acros che imputava la falla esclusivamente ad iTunes ma poi si è scoperto essere un problema nettamente più diffuso a tutti i software che hanno interazioni con le librerie dinamiche DLL. Microsoft ha rilasciato un Security Advisory (2269637) lo scorso 23 Agosto 2010 confermando l'esistenza della vulnerabilità e specifica che sono soggette all'exploit esclusivamente le applicazioni che non caricano le librerie in modalità sicura secondo le loro linee guide disponibili sul portale di MSDN.

DLL Load Hijacking

Un'applicazione si può definire affetta dal DLL Load Hijacking quando carica librerie DLL estranee a quelle originali, considerando l'apertura di un mp3 tramite il player multimediale iTunes, quest'ultimo dovrebbe caricare esclusivamente le proprie librerie autentiche ma realmente carica anche librerie estranee contenute nella cartella del mp3.
Un malintenzionato pertanto potrebbe sfruttare l'exploit del DLL Hijacking inserendo una libreria DLL malevola all'interno di una cartella condivisa, un archivio o un torrent assieme a uno o più file innocui; all'apertura di un file innocuo si caricherà anche la libreria DLL che provvederà ad infettare il PC della vittima.
Ipotizziamo di distribuire sul canale Torrent l'album musicale più richiesto del momento, l'archivio non conterrà esclusivamente l'album ma anche la DLL infetta. L'utente una volta completato il download dell'album provvederà ad estrarlo ed ad ascoltare l'album, ed ecco che viene caricata la DLL infetta contenente sicuramente un Malware/Trojan.
Ovviamente non tutte le applicazioni sono affette da tale vulnerabilità e neanche l'attaccante può sapere quale software usa il bersaglio per riprodurre i file multimediali, pertanto userà la tecnica del Multi Exploiting DLL Hijack ovvero l'inserimento all'interno dell'archivio di più DLL in fette (ognuna per ogni lettore multimediale) aumentando le probabilità di eseguire l'attacco.
Il team di metasploit ha già sviluppato il kit DLLHijackAuditKit in grado di analizzare ogni applicazione e determinare se è affetta da tale vulnerabilità, ed in pochi giorni si è scoperto che più di 200 applicazioni che interagiscono con file esterni (mp3, avi, html, png, dwg, tiff, ecc ecc) sono affette da tale vulnerabilità.
Pertanto vi consiglio massima attenzione nell'aprire contenuti remoti e rendere sempre visibili i file nascosti, se visualizzate un file DLL assieme ad altri file "sicuri" provvedete ad analizzarlo, potrebbe essere un tentativo di attacco al vostro terminale.

metasploit

Tuesday, August 24, 2010

Better, Faster, Stronger: DLLHijackAuditKit v2

Due to an overwhelming amount of interest in the initial DLLHijackAuditKit released on Monday, I rewrote the tool to use native JScript, automatically kill spawned processes, reduce the memory usage by ProcMon, and automatically validate every result from the CSV log. The result is DLLHijackAuditKit v2. This kit greatly speeds up the identification process for vulnerable applications. An extremely simple HOWTO:

1. Download the DLLHijackAuditKit v2 and extract it into a local directory on the system you would like to test.

2. Browse to this directory and launch **01_StartAudit.bat** as an Administrator. The Administrator bit is important, as it will allow the script to kill background services that are spawned by the handlers and prevent UAC popups.

3. After the audit script completes (15-30 minutes), switch to the Process Monitor window, and access File->Save from the menu. Save the resulting log in CSV format to the local directory with the name "Logfile.CSV".

4. Launch **02_Analyze.bat** as an Administrator. This will scan through the CSV log, build test cases for each potential vulnerability, try them, and automatically create a proof-of-concept within the Exploits directory should they succeed.

5. Identify the affected vendor for each generated proof-of-concept and ask them nicely to fix their application. Send them the calc.exe-launching PoC if necessary.

Thanks again to everyone who provided feedback (positive or negative) on the original tool, especially Rob Fuller, who let me forkbomb his system in the process of testing the new kit.

Generated by hdm at 11:44:00 PM

Products

Metasploit Express
Rapid7 NeXpose

Links

The Metasploit Project
The Uninformed Journal
WarVOX: Wardialing++
Decloak: DeAnonymized

Blogs

Rapid7 Security
Attack Research
RISE Security

Archive

▼ 2010 (23)
Sep 2010 (1)

di Spirit
redazione@hackerjournal.it

DEFACCIARE CHE PASSIONE

HACKING

Il defacciamento di un sito è spesso solo un atto dimostrativo, altre volte nasconde realtà più complesse.

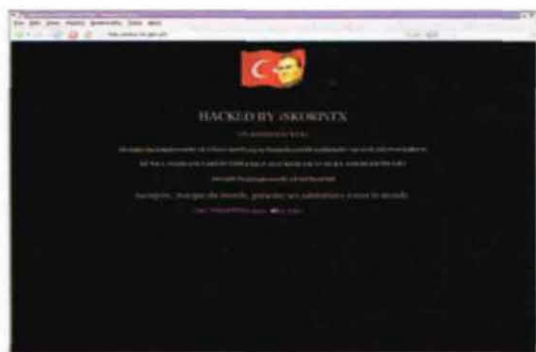
Defacciare è una parola che ha cominciato ad assumere una certa popolarità intorno alla fine degli anni '90, quando il fenomeno internet ha cominciato a crescere e, sebbene sia una terminologia di strettissima attualità, fatica ancora a trovare uno spazio nei dizionari come neologismo. Con Defacciamento si intende, in termini davvero generali, spesso la semplice e sola sostituzione della pagina di index di un sito con un'altra di contenuti diversi caricata proprio dal "defacciatore". I contenuti della nuova pagina di indice caricata possono essere davvero molteplici, dipende tutto sommato da quali sono le intenzioni di chi porta questo genere di attacchi, si va dalla schermata burlesca fine a se stessa a veri e propri proclami e atti di denuncia.
Ma questa è evidentemente solo la punta dell'iceberg. Il fenomeno del Defacement ha implicazioni spesso più profonde. I siti web che subiscono pesanti attacchi di defacement sono altre volte trasformati in veri e propri nodi di botnet, backdoor e account di shell venduti al mercato nero: questo è ciò che quotidianamente avviene su Internet. Il presente articolo analizza le modalità e gli strumenti utilizzati dai "defacer" per condurre i loro attacchi nei confronti dei siti web, fornendo al contempo preziosi consigli per evitare di divenire noi stessi facile bersaglio di tali malintenzionati.

IL "DEFACCIATORE"

Il defacer, in genere, non presta mai particolare attenzione al tipo di sito web da sottoporre ad attacco; il suo scopo principale rimane quello di individuare e sfruttare al meglio le vulnerabilità presenti in certi server, per poi modificare il contenuto o l'aspetto visivo dei siti web violati, oppure lasciare tracce tangibili della "cortese" visita effettuata, caricando magari nel server compromesso un file che evidenzi l'azione di defacing appena compiuta. In realtà nessuno sa spiegare concretamente perché i defacer agiscano in tal modo, visto che la loro losca attività non sembra produrre alcun evidente profitto in termini pecuniari. Esplorando tuttavia gli appositi archivi online che raccolgono e catalogano gli innumerevoli "exploit" realizzati dai defacer, ci rendiamo subito conto che è in atto ormai da tempo una strenua competizione tra i vari gruppi di defacer operanti in Rete, al fine di realizzare le migliori performance in termini di danneggiamento del maggior numero possibile di siti web. Sebbene i media continuino a definire semplicemente come "hacker" le persone che compiono tali atti, vorrei in ogni caso precisare che i "veri" hacker non conducono mai attacchi casuali nei confronti dei siti Internet, ma sfruttano al meglio le conoscenze tecniche acquisite per realizzare azioni di hacking ben mirate. Gli hacker cercano inoltre di proposito di evitare che i proprietari dei siti compromessi possano rendersi in qualche modo conto dell'attacco subito; al contrario, essi fanno sempre tutto il possibile per nascondere o cancellare ogni traccia o evidenza dell'attacco portato.

DEFACEMENT

Gli attacchi eseguiti dai defacer sono comunemente definiti con il termine di "defacement"; in

Spesso l'attività di defacciamento ha scopi puramente divulgativi ideologici: serve in sostanza per comunicare un messaggio, anche estremo.

Internet esiste tutta una serie di siti web che fungono da veri e propri archivi (peraltro molto estesi) delle azioni di defacement compiute: come riferito sopra, esiste altresì un'ampia comunità di defacer, i cui gruppi e membri sono in perenne concorrenza tra loro per stabilire chi sia in grado di craccare e danneggiare il maggior numero possibile di siti web. Tali archivi sono pubblicamente accessibili, il che significa che ogni gruppo di defacer può ad ogni momento verificare quanti "punti" esso sia riuscito a collezionare in classifica, tenendo al contempo sott'occhio i "successi" ottenuti dai gruppi rivali.I defacer non applicano alcun criterio di selettività nei confronti dei loro obiettivi; nella maggior parte dei casi essi si avvalgono semplicemente di tool automatizzati preposti ad individuare i server vulnerabili, per poi sfruttare questi ultimi in maniera ugualmente del tutto "automatica". L'exploit carica automaticamente sul server compromesso una backdoor la cui funzione è, ad esempio, quella di fornire l'accesso shell a tale server. Il defacer può ovviamente lanciare ulteriori attacchi tramite la suddetta backdoor, per cercare di aumentare i privilegi grazie agli exploit del kernel locale, o magari segnalare il server compromesso ad un apposito archivio di defacement. Simili backdoor vengono vendute anche sul mercato nero della cybercriminalità, consentendo in tal modo ai loro acquirenti di poter ad esempio trasformare un server violato in un vero e proprio nodo di una rete DdoS, o magari di utilizzarlo in qualità di host per inoltrare e-mail di spam.
Una volta che l'attacco è andato in porto, l'azione di defacement compiuta viene automaticamente segnalata ad un archivio online.

GLI STRUMENTI

I defacer si avvalgono innanzitutto di appositi scanner per individuare i server vulnerabili da sfruttare; una volta completato il processo di scansione ed identificati i server-vittima, i malintenzionati provvedono a caricare all'interno di questi ultimi speciali backdoor in grado di fornire loro preziose *informazioni riguardo alle macchine* infettate e di svolgere, al tempo stesso, la funzione di scanner aggiuntivi. Nella maggior parte dei casi, gli exploit utilizzati dai defacer sono pubblicamente disponibili, anziché essere del tipo "zero-day". Per identificare i server vulnerabili i defacer ricorrono spesso all'utilizzo delle "Google Dorks": si tratta di particolari query di ricerca, le quali possono essere ad esempio eseguite per ottenere determinati risultati riguardanti tutti i siti web in cui risulta attiva una specifica versione di una certa applicazione. In alcuni casi, è la backdoor stessa utilizzata dai defacer a generare il download di speciali database contenenti particolari Google Dorks, divenendo in tal modo una sorta di nodo di scansione dedicato alla ricerca di nuovi server vulnerabili.
Gli strumenti utilizzati dai defacer per individuare nuovi server vulnerabili verificano in primo luogo la presenza di due tipi di vulnerabilità: le vulnerabilità per file remoti e quelle di tipo Local File Include. Riportiamo qui di seguito un elenco parziale di tali strumenti, che risultano peraltro essere del tutto gratuiti e pubblicamente disponibili:

LFI intruder
VopCrew IJO Scanner v1.2
Single LFI vulnerable scanner
SCT SQL SCANNER
Priv8 RFI SCANNER v3.0
PITBULL RFI-LFI SCANNER
Osirys SQL RFI LFI SCANNER
FeeLCoMz RFI Scanner Bot v5.0 By FaTaLisTiCz_Fx

Come accennato in precedenza, una volta individuato il server vulnerabile, i defacer provvedono a generare il download di un'apposita backdoor all'interno di tale server. Le backdoor presentano una vasta gamma di funzionalità, ma la maggior parte di esse possiede dei metodi specifici per bypassare le funzioni di sicurezza PHP, rubare informazioni, leggere e modificare i file, accedere a database SQL, craccare password, eseguire comandi arbitrari e modificare i privilegi. Nel corso della mia ricerca ho rilevato oltre un centinaio di differenti backdoor e shell PHP;

Spirit
redazione@ha

pare, tuttavia, che la maggior parte delle backdoor individuate in sostanza utilizzi gli stessi tipi di base, ovverosia:

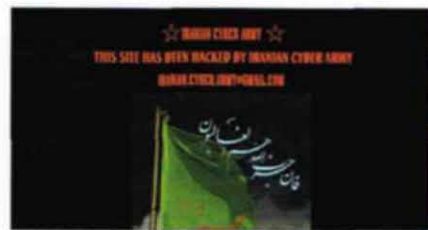
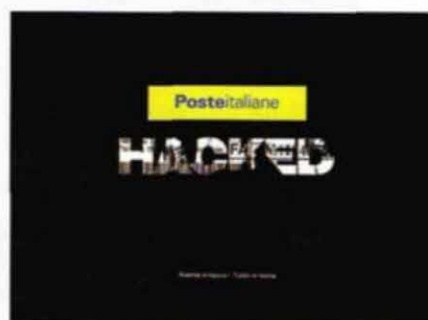
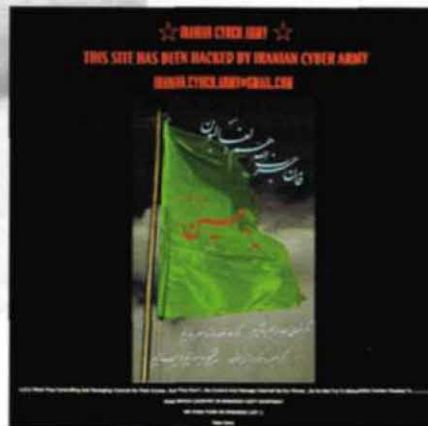
```
r57
c99
Locus7Shell
```

Le modalità di cui si avvalgono le backdoor per cercare di modificare i privilegi consistono principalmente nell'utilizzo di "auto-rooters" o nel tentativo di estrarre le password dai file di configurazione custoditi all'interno del server compromesso. I cosiddetti "auto-rooters" altro non sono che script di shell che provvedono a generare nel server il download di uno specifico kit di exploit, composto da exploit precompilati già pronti per essere eseguiti. Lo script di shell analizzerà in seguito la macchina compromessa, al fine di determinare quali sono gli exploit da eseguire; verrà infine lanciata l'esecuzione di questi ultimi. Se l'exploit riesce nell'opera di modificare con successo i privilegi, avrà poi luogo l'installazione di un'altra backdoor o di un rootkit. Gli "auto-rooters" vengono offerti in Rete da numerosi siti.

SOLUZIONI

no dei problemi più grandi nel combattere gli attacchi di defacement è che i defacer non sfruttano solamente vulnerabilità tecniche ma anche l'ignoranza di numerosi operatori. In effetti, la maggior parte delle persone che lavorano con i server web non comprende ancora pienamente l'importanza di avere un sistema costantemente aggiornato e dotato di tutte le ultime patch disponibili. L'installazione delle patch via via rilasciate dai produttori di software, indipendentemente dalla fondamentale importanza che essa riveste, risulta oltretutto un'operazione piuttosto semplice da eseguire; nonostante ciò, una delle questioni più comuni tutt'oggi legate alla sicurezza online è ancora rappresentata dal non mantenimento di adeguati standard di aggiornamento per il sistema informatico utilizzato. Le società e le organizzazioni spesso spendono molto tempo ed energie per spiegare al proprio personale IT come funzionano le iniezioni SQL ed i buffer overflow, e come possono essere sfruttate per un attacco, quando sarebbe invece molto più utile ed opportuno concentrarsi nel garantire che i sistemi siano completamente aggiornati e configurati in maniera appropriata.

L'OS

Un'altra questione di fondamentale importanza è rappresentata dal fatto che gli amministratori danno spesso per scontato che il sistema operativo Linux/Unix sia più sicuro di Windows; in tal modo, essi non provvedono a rafforzare adeguatamente le misure di sicurezza adottate e ad aggiornare le configurazioni utilizzate. Una corretta configurazione del sistema può in effetti risultare determinante per eliminare certi tipi di exploit. Ad esempio, molti degli exploit menzionati nel presente articolo sfruttano, in sostanza, vulnerabilità del tipo "File Include", le quali consentono al defacer di introdurre all'interno del server compromesso qualsiasi tipo di file arbitrario; in alcuni casi tali file possono provenire anche da siti web esterni. Per proteggersi efficacemente nei confronti di simili attacchi sarebbe pertanto sufficiente specificare la directory dalla quale una determinata applicazione web od un sito web risultano autorizzati ad effettuare l'inclusione di file all'interno del server.

Recentemente l'organizzazione Iranian Cyber Army ha "defacciato" l'home page del più importante motore di ricerca cinese, ovvero Baidu.com, che per inciso, supera come utenti di gran lunga Google.

Il defacciamento, a scopo puramente dimostrativo, del sito delle Poste Italiane. Per fortuna l'intento era quello di stupire più che di colpire: buon per gli amministratori di sistema che hanno potuto porvi rimedio.

Sempre l'Iranian Cyber Army si era preoccupato di portare, qualche tempo fa, un attacco a Twitter.

di Marco Scialdone
http://scialdone.blogspot.com

QUALE FUTURO PER LA NET NEUTRALITY?

IN RETE

LA PROPOSTA DI CREARE UNA RETE REALMENTE APERTA È UN'UTOPIA O UN PRIMO PASSO VERSO UN FUTURO DI LIBERTÀ CONDIVISE?

Quella appena trascorsa è stata un'estate all'insegna delle discussioni sul futuro della Net Neutrality. A tenere banco ancora una volta Google che, insieme a Verizon, ha annunciato la sua "proposta per una Rete aperta" (disponibile in italiano su http://googlepolicyeurope.blogspot.com/2010/08/joint-policy-proposal-for-open-internet.html). Nel preambolo le due aziende hanno inteso formalizzare i principi che intendono sostenere in merito alla complessa questione della net neutrality, avendo come obiettivo, da un lato quello di salvaguardare il diritto degli utenti di essere liberi di scegliere quali contenuti, applicazioni o dispositivi usare, dall'altro quello di spronare il governo statunitense a continuare negli investimenti a supporto dell'infrastruttura di banda larga. In linea teorica, dunque, le due aziende hanno manifestato la loro incondizionata adesione all'idea di una Rete neutrale.

Scorrendo il testo, però, ci si rende conto che si è lontani da quell'idea di neutralità secondo la quale non possono esistere discriminazioni tra i contenuti che viaggiano all'interno di una rete di computer. Google e Verizon non negano la possibilità del ricorso a tecniche di network management, chiedendo soltanto che ciò avvenga in maniera trasparente rispetto ai fornitori di applicazioni e contenuti. Inoltre la proposta, se tradotta in regolamentazione vigente, autorizzerebbe gli operatori a fare servizi a valore aggiunto, distinti da Internet, per i quali non varrebbe principio di neutralità, neppure nella versione "soft" da loro suggerita. Per non parlare del traffico wireless che, secondo Google e Verizon, dovrebbe essere tenuto esente da ogni regolamentazione in materia. E' evidente come sul tema della neutralità della rete si confrontino da tempo due diverse concezioni: una che mostra maggiore attenzione all'architettura della Rete quale strumento di tutela della concorrenza, dell'innovazione, della libertà di espressione, l'altra più attenta alle esigenze del mercato e, dunque, allergica ad una regolamentazione prettamente giuridica di fenomeni che dovrebbero trovare nella pura competizione il rimedio naturale ai potenziali abusi.

Quale delle due concezioni finirà con il prevalere è presto per dirlo. Tuttavia, l'Internet che conosciamo ha prodotto innovazione, confronto, apertura e lo ha fatto in virtù di un'architettura che gli impediva di discriminare. Fosse anche solo per questo, sulla Net Neutrality, come ha sostenuto Lawrence Lessig, converrebbe essere conservatori.

di NoeXKuzE
redazione@hackerjournal.it

PS3 BUCATA!

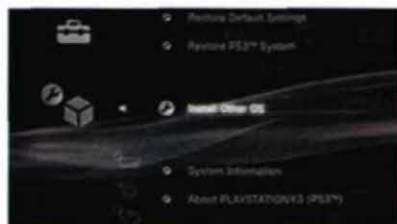
HACKING

QUATTRO ANNI DOPO IL SUO LANCIO ANCHE LA PS3 È STATA RESA PIÙ "DOCILE".

Qualcuno pensa che la SONY abbia agito in modo davvero poco lungimirante perché le notizie che circolano in rete da qualche settimana somigliano più a una guerra senza quartiere che a semplici scoperte di falle software. Evidentemente la politica adottata ha fatto inquietare più di una persona!

UN PO' DI STORIA

A fine novembre del 2006 la SONY ha lanciato sul mercato un gioiello tecnologico chiamato per puro marketing Playstation 3. Di fatto non si trattava di una semplice revisione della linea Playstation, ma di un progetto fatto da zero che permetteva di avere a un costo decisamente basso una potenza di calcolo (parallelo) per il tempo incredibile. Tanto che il processore stesso (Cell) fu oggetto di interesse fin da subito da parte dei militari, statunitensi in testa.

Tra le novità più interessanti veniva introdotto il primo lettore Blu-Ray a bordo di una consolle domestica, con tanto di uscita HDMI 1080p e veniva fornito il collegamento alla rete tramite porta ethernet e Wi-Fi. In alcuni modelli venne reso disponibile anche un lettore di schede di memoria, mentre in tutte le versioni erano presenti almeno due porte usb cui collegare hard-disk esterni e anche tastiere.

Con gli ultimi test sulle ps3 aperte si riesce a ripristinare OtherOS!

Non ultimo per importanza, venne pubblicizzato fin da subito il supporto noto come OtherOS, dal nome dell'opzione che era disponibile fino al primo aprile di quest'anno all'avvio della macchina. Un'opzione pensata esclusivamente per far girare anche Linux sulla PS3.

Ma ad aprile di quest'anno la SONY, tramite un semplice aggiornamento "obbligatorio" del firmware, ha rimosso tale caratteristica venduta fino al giorno prima. La motivazione ufficiale è legata al fatto di voler garantire la sicurezza dell'utente, ma nei fatti hanno voluto togliere una caratteristica peculiare che ha certamente contribuito a far crescere l'interesse verso la PS3 anche grazie alla comunità opensource, che si è vista tradita.

Ufficiosamente la rimozione di OtherOS è legata all'hack realizzato da GeoHoz lo scorso gennaio, che permette di entrare in una modalità di debug proprio a partire da linux. Possibilità che dovevano rimanere ristrette agli addetti ai lavori, un rischio troppo grande per il management di SONY.

IL CONTRATTACCO

In pieno agosto è scoppiata una bomba software: un'azienda australiana ha iniziato a pubblicizzare una chiavetta usb con la funzione dichiarata di realizzare l'exploit di GeoHoz senza intaccare minimamente la garanzia della consolle e sfruttando il firmware 3.41 bloccato di SONY. Il dispositivo è stato chiamato PS3 JailBreak e inizialmente veniva venduto a 160 dollari. Insieme alla chiavetta veniva poi consegnato Backup Manager, un software in grado di copiare i giochi dai super-protetti Blu-Ray a un hard-disk. La notizia era colossale: dopo quattro anni dal rilascio, quella che sembrava fino ad allora una consolle inviolabile apriva uno spiraglio verso la pirateria dei giochi!

Una piccola chiavetta usb ha scatenato il panico in casa SONY tanto che in alcuni stati al momento è illegale persino venderle.

Questo software è in grado di copiare l'intero contenuto di un gioco presente in un disco Blu-Ray, su un hard-disk che può essere quello interno alla consolle o uno esterno connesso via usb. Una volta compiuta la copia, il gioco viene riconosciuto al pari del disco e non è più necessario averlo nel lettore Blu-Ray (questo grazie a un recente fork del codice di PS Groove realizzato da Hermes). A seguito del rilascio di Backup Manager, è stata realizzata una classifica di tutti i giochi che possono essere copiati e che risultano funzionanti (vedi https://spreadsheets.google.com/lv?key=tqjzdwQGOhsHl_KH0KiEC3w&toomany=true).

Il tam-tam nella rete è stato immediato e distributori pronti a rivendere tale chiavetta sono iniziati a comparire in tutto il mondo, mentre dalla Cina veniva annunciato il primo clone X3 JailBreak, cui ne seguono nuovi (funzionanti o meno) quotidianamente.

La SONY ha iniziato ad agire per vie legali riuscendo a bloccare la vendita ufficiale delle chiavette, che però nel frattempo hanno iniziato a circolare comunque in tutto il mondo, finché le stesse non sono state analizzate e reversate da sviluppatori opensource che hanno studiato l'exploit hardware e hanno rilasciato un codice opensource chiamato PS Groove (http://github.com/psgroove/psgroove) in grado di realizzare la stessa cosa ma su hardware di basso costo (intorno ai 35 dollari). La mossa decisiva è stata ancora una volta il rilascio del codice in opensource: grazie a questa mossa, sono aumentati a dismisura i modi per realizzare l'exploit sfruttando un iPhone, una PSP, una calcolatrice TI-84 e molti altri device.

Ai primi di settembre la SONY ha rilasciato una versione affrettata del firmware, la 3.42, in grado di tappare la falla (affrettato perché molti utenti hanno lamentato blocchi e malfunzionamenti sulle consolle aggiornate). Inoltre tramite un controllo remoto, possibile sulle consolle che si collegano alla rete di SONY (PSN), sono iniziati anche i ban degli utenti che presentavano installato Backup Manager e impedendo il collegamento degli utenti con firmware precedente al 3.42.

Come contro attacchi, sono stati rilasciati dei tool che permettevano di collegarsi a PSN tramite un PC che simulava di essere una PS3 con firmware 3.42 (anch'esso presto bloccato) e nuove versioni del Backup Manager che tentano di impedire il ban tramite il proprio camuffamento (si identifica come un gioco selezionabile dall'utente), ma chiaramente è una misura temporanea dato che SONY cerca di costringere tutti quanti a passare alla versione bloccata del firmware.

Mentre scriviamo è stata rilasciata la versione 3.50 che risolve i bug della 3.42 e introduce il supporto 3D Blu-Ray (del tutto inutile se non si dispone di un TV 3D). Ovviamente anche questa è una mossa strategica che tenta di convincere la maggior parte degli utenti ad aggiornarsi il prima possibile per chiudere le falle scoperte. In questo momento esistono quindi PS3 con firmware 3.41 e bucate tramite jail-break (magari di utenti non interessati a PSN) e PS3 regolarmente aggiornate per le quali non è escluso che si possa comunque realizzare comunque il jail-break con i prossimi tool.

LE NOVITÀ

Grazie all'eco mediatico del jail-break la comunità degli utenti

in tutto il mondo è in pieno fermento e hacker e sviluppatori rilasciano ormai quotidianamente nuovi tool e i primi software opensource per PS3. Per citare solo i principali, dopo PS Groove, sono stati infatti rilasciati un File Manager (Comgenie's Awesome File Manager, http://www.ps3hax.net/2010/09/comgenies-awesome-filemanager-released/) e un FTP Server (PS3 FTP Server, http://www.ps3-hacks.com/file/92). Questi tool di base hanno permesso di realizzare nuovi hack e nuovi software come:
xRegistryEditor, un tool che permette di modificare il registro di XMB (il sistema operativo delle playstation)
una nuova release di PS Groove che permette di effettuare il dump del kernel in funzione sulla propria consolle (tramite comandi di peek e poke)
Jaicrab USB Firm Loader e Kammy, due tool che permettono di giocare con il firmware ufficiale della PS3.

PS3 FTP SERVER

Questo strumento fornisce un coltellino svizzero per dissezionare con calma il filesystem della PS3. Navigando da un PC connesso in rete alla PS3 è infatti possibile vedere tranquillamente la flash, l'hard-disk interno, le porte usb e l'unità Blu-Ray e si ha accesso alle unità con permessi differenti. In particolare:
pieno accesso in lettura e scrittura all'hard-disk (dev_hdd0), inclusi i dati di gioco
pieno accesso in lettura e scrittura in dev_flash2 e dev_flash3
pieno accesso in lettura e scrittura ai dispositivi connessi alla usb
pieno accesso in lettura in dev_flash e dev_bdvd

COMGENIE'S AWESOME FILE MANAGER

Questo è un vero e proprio file manager che gira direttamente sulla PS3 e permette di copiare i file tra la PS3 (e le sue periferiche) e ad esempio un hard-disk esterno.

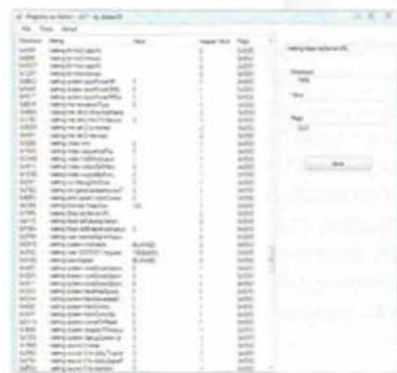
Tra le caratteristiche principali:
la possibilità di copiare file più grandi di 4Gb (e fino a 15Gb, perché oltre possono corrompersi) spezzandoli in più blocchi e rifondendoli nel processo inverso
copiare e cancellare intere cartelle

Il primo file manager per PS3 costruito in casa permette di scoprire come funziona XMB e dove vengono salvate le informazioni

Un gioco splittato non può essere lanciato, tuttavia si può comunque realizzare un backup che permette di ricostruirlo sulla consolle. Un gioco che invece entra in 4Gb può essere lanciato direttamente dall'hard-disk esterno.

XREGISTRYEDITOR

L'autore di questa scoperta si fa chiamare SKFU (http://streetskaterfu.blogspot.com/2010/09/ps3-registry-research.html). SKFU utilizzando l'ftp manager ha indagato sul file xRegistry.sys scoprendo che diverse informazioni vengono memorizzate in binario al suo interno. Grazie a xRegistryEditor (http://stoker25.com/?p=96) realizzato da stoker25, è possibile modificare (per ora) le seguenti informazioni:

il tuo username locale
la lingua della consolle (es. "eng" per inglese)
il nome della tua consolle
l' URL che punta ai file che mantengono le informazioni online (board)
il seriale dell'hard-disk interno
il nome di board
il tuo nome utente su PSN e la password
la tua chiave di rete WIFI
il tuo IP locale
il tuo PSID
il percorso locale alle tue immagini

Con l'ultima release del xRegistryEditor si possono tenere sotto controllo i checksum e i valori di tutte le variabili interne di XMB

JAICRAB USB FIRM LOADER

Avendo a disposizione una PS3 bucata e il PS3 FTP server, è possibile grazie a questo tool

effettuare il dump di /dev_flash e installarlo su una penna usb da cui poi fare il boot. Infatti il programma simula che la penna sia una periferica dev_flash che viene quindi gestita come tale da XMB. Questo apre la porta alla possibilità di creare firmware custom da lanciare sulla propria consolle in modo controllato, visto che funziona su una penna.

L'unica accortezza da osservare è quella di non effettuare alcun update da internet mentre stiamo utilizzando il firmware caricato dalla penna, perché la scrittura avverrebbe comunque sulla flash fisica nella consolle.

Per installarlo, una volta scaricato (http://www.ps3hax.net/downloads.php?do=file&id=422) va prima copiato tramite ftp server tutto il contenuto di /dev_flash sulla penna USB e va aggiunto il pacchetto JaiC_USB_FIRM_LOADER.pkg sempre nella root. Si installa il tool e si inserisce la penna nella PS3. A questo punto va lanciato USB FIRM LOADER e dopo qualche secondo XMB segnalerà che è presente una nuova flash.

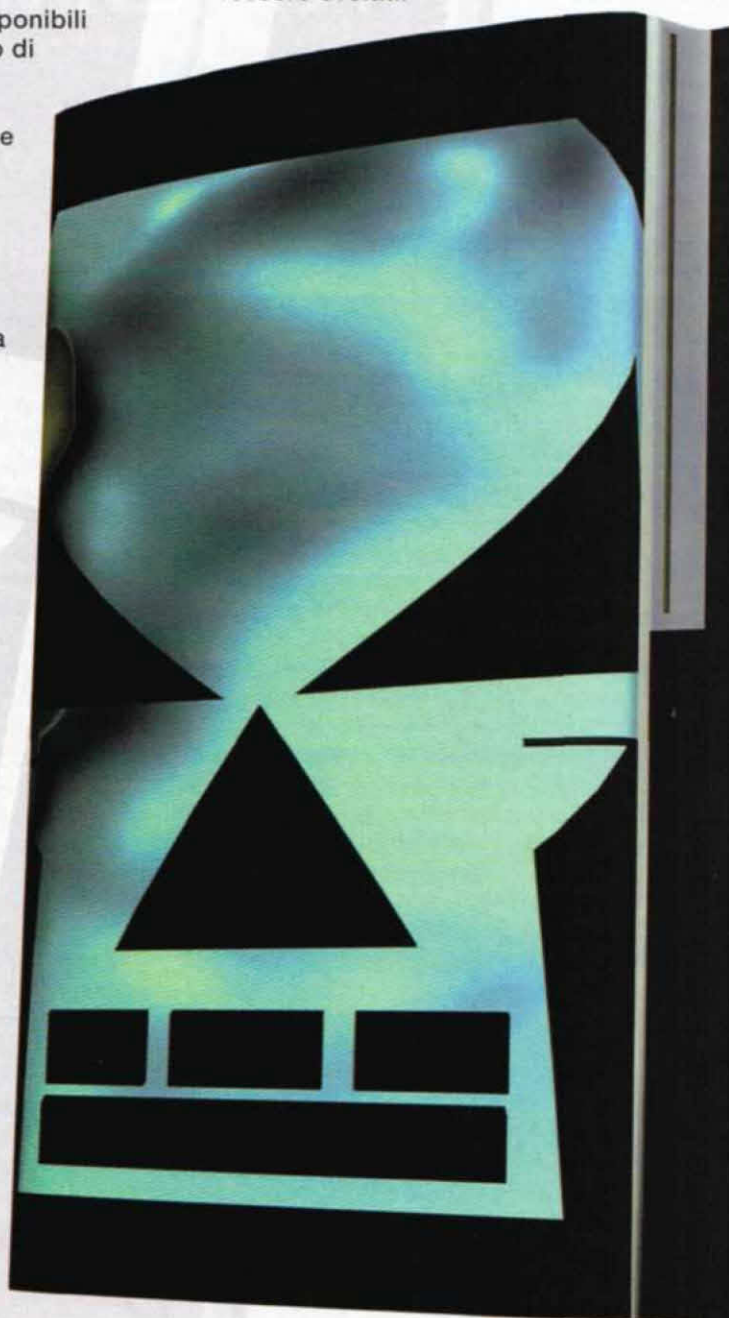
KAMMY

Kammy (http://www.ps3news.com/PS3-Hacks/kammy-userland-ps3-lv2-gameos-patches-for-psgroove-arrive/) rappresenta una novità addirittura superiore al tool di Jaicrab perché permette di gestire delle patch da applicare direttamente al firmware. Viene rilasciato in sorgenti compilabili con il gcc e si innesta direttamente sul lavoro egregio svolto con PSGroove.

IL FUTURO

Le ultime notizie disponibili parlano del tentativo di riattivare l'opzione OtherOS sulle PS3 aggiornate e le cause in corso (intentate da utenti che avevano acquistato la consolle prima dello scorso primo aprile) potrebbero costringere la stessa SONY a ripristinare la funzionalità tolta. Lo stesso esercito americano utilizza cluster di PS3 con linux a bordo che con i recenti aggiornamenti rendono la manutenzione impossibile.

La guerra per il controllo della consolle è cominciata ed è difficile dire ora se avrà termine oppure no. La PS4 è lontana dal suo rilascio e nel frattempo la PS3 viene analizzata come non era mai accaduto negli ultimi quattro anni.

Alcuni si chiedono se tutto ciò che è successo, sarebbe accaduto comunque dato che prima o poi tutte le consolle vengono violate e la pirateria tende a prendere il sopravvento. Forse sì, tuttavia l'aver tolto alla comunità uno strumento troppo appetibile come linux sulla PS3 ha probabilmente acceso l'animo di persone maliziose e puntato i riflettori su argomenti che per la SONY sarebbe stato più opportuno non fossero svelati.

di Giovanni Federico - info@giovannifederico.net
Fabio `BlackLight` Manganiello - blacklight86@gmail

PARTE VII

CORSO DI PROGRAMMAZIONE IN C

LINGUAGGI

LA SETTIMA PARTE DEL CORSO È DEDICATA A STRINGHE E FILE.

PROGRAMMAZIONE MULTIPROCESSO E MULTITHREAD

Negli anni '80 il grosso vantaggio dei sistemi Unix-based rispetto a sistemi come DOS consisteva nel fatto che questi erano progettati per essere multitasking, ovvero eseguire più di un processo per volta. Un programma al momento dell'esecuzione diventa un processo residente in memoria centrale ed eseguito dalla CPU, e finora abbiamo esaminato in questo corso programmi di questo tipo (ovvero un singolo programma a cui è associato un singolo processo in esecuzione). Si pensi tuttavia a sistemi come le prime versioni di *MS-DOS (per chi ha buona* memoria): se si esegue la copia di una directory attraverso il comando copy il sistema rimane "congelato" fino a quando la copia non è terminata (con ovvi disagi nel caso in cui si debbano copiare molti file), in quanto il processo copy monopolizza la CPU fino alla sua terminazione. Su un sistema operativo moderno possiamo invece, allo stesso tempo, tenere aperti il nostro browser, un media player e un terminale. E anche all'interno dello stesso programma possono essere presenti più processi che operano in maniera concorrente (si pensi a un browser moderno, contenente un processo per la gestione dell'interfaccia grafica, uno per il parsing dei contenuti HTML che arrivano, uno per la gestione delle connessioni di rete, uno per la gestione delle estensioni, e così via). Si tratta di un passaggio da un paradigma monoprocesso a uno multiprocesso o multitasking. In realtà, a meno che non si parli di sistemi multicore (con tutte le difficoltà implementative annesse e connesse per la comunicazione e la sincronizzazione fra i diversi core), la CPU rimane una, e fisicamente non è possibile che diversi processi siano in esecuzione allo stesso tempo su una sola CPU, in quanto lo stesso set di registri, gli stessi bus di comunicazione e le stesse risorse hardware non possono essere in uso allo stesso tempo da diversi processi. *L'impressione che i processi siano attivi* contemporaneamente è quindi un'illusione agli occhi dell'utente, mentre in realtà in ogni istante c'è solo un processo in esecuzione sulla CPU. Semplicemente la CPU riserva a ogni processo un time slice per cui quest'ultimo può occupare la CPU, scaduto il quale torna in coda e il turno passa al processo successivo. Il sistema fa ciò attraverso algoritmi di scheduling, il più semplice dei quali è il round robin (da immaginare come una coda circolare in cui ogni processo può occupare la CPU per t nanosecondi prima di tornare in coda), per poi arrivare ad algoritmi più complessi implementati sulle CPU moderne (code dinamiche, code a priorità, code che possono premiare o penalizzare processi in base al tempo di CPU che richiedono, e così via). L'argomento è piuttosto complesso e per ovvi motivi di spazio non può essere trattato qui, ma per anni la complessità dello scheduling dei processi in un sistema multitasking è stata una barriera implementativa notevole sia a livello hardware, sia a livello di complessità del software, rendendo tali sistemi di difficile realizzazione a costi contenuti, ed è ancora oggi un problema per dispositivi embedded o con limitata potenza di calcolo. Inoltre la buona programmazione multiprocesso è un tema estremamente complesso, in quanto la sincronizzazione fra diversi processi concorrenti in un software molto complesso è tutt'altro che banale, e richiede abilità di "visione d'assieme" ed esperienza non indifferenti per il programmatore.

PROGRAMMAZIONE MULTIPROCESSO

Un sistema Unix-based è fortemente multiprocesso. In particolare, al momento dello startup, viene avviato un processo chiamato init che ha priorità massima. Tale processo genera diversi processi figli,

ognuno identificato da un proprio PID. Per l'autenticazione da riga di comando dello username viene avviato il processo login, e per l'autenticazione della password viene sfruttato il processo getty, e dopo un'autenticazione completata con successo viene avviata una shell (nel caso di tty) che è un processo figlio di init. E a sua volta da questa shell possiamo dare diversi comandi, che saranno tutti processi figli della shell che li ha richiamati. Nel caso di un'ambiente grafico, ogni applicazione grafica avviata è generalmente un processo figlio del desktop environment o del window manager stesso. Per creare all'interno di un programma C un nuovo processo si usa la primitiva fork(). Tale primitiva non prende nessun parametro e ritorna:

- 0 nel caso in cui siamo nel codice del processo figlio;
- un valore > 0 (corrispondente al PID del figlio generato) se siamo nel codice del processo padre;
- -1 se il processo figlio, per qualsiasi motivo, non è stato generato.

Esempio:

```
#include <stdio.h>
#include <unistd.h>
#include <sys/wait.h>

int main ( int argc, char **argv )
{
        int pid, status;

        printf ( "Sono il processo padre e il mio PID e' %d\n", getpid() );

        /* Genero un nuovo processo */
        pid = fork();

        if ( pid == -1 ) {
                printf ( "ERRORE! Non posso creare un nuovo processo :(\n" );
                exit ( 1 );
        } else if ( pid == 0 ) {
                printf ( "Sono il processo figlio di %d, il mio PID e' %d\n",
                        getppid(), getpid() );
                exit ( 0 );
        } else {
                printf ( "Sono il processo padre e ho generato "
                        "con successo un figlio\n" );
                while (( pid = wait ( &status )) > 0 );
                status = ( status & 0xFF ) >> 8;

                printf ( "Mio figlio e' morto e ha ritornato il valore %d\n",
                        status );
        }

        return 0;
}
```

Si noti l'uso delle funzioni getpid() e getppid() per ottenere il process identifier (PID) rispettivamente del processo corrente e del processo padre. Il processo padre quindi esegue le operazioni che deve eseguire dopo aver lanciato il figlio o i figli, per fare infine un ciclo di wait in attesa che tutti i processi figli terminino prima di continuare. La wait ritorna un valore maggiore di zero fintanto che ci sono processi figli ancora in esecuzione, quindi scrivere while ((pid = wait (&status)) > 0); equivale a dire "finché esiste almeno un processo in esecuzione (il cui PID viene salvato nella variabile pid), attendi". La variabile status conterrà, nel byte più significativo, il valore di ritorno del processo figlio. Per leggere tale valore è quindi necessario fare un AND binario e uno shift. Se non mi interessasse il valore di ritorno del figlio potrei richiamare la wait attraverso come wait ((int*) 0). È indispensabile che il processo padre si accerti che tutti i suoi processi figli sono terminati prima di terminare a sua volta. Quando un processo figlio termina deve infatti ritornare al chiamante (ovvero il processo padre) il suo status di uscita per confermare la sua terminazione, e se il padre è già terminato il processo passa in status zombie, ovvero terminato ma in attesa di ritornare un valore al padre. Tali processi vengono generalmente adottati dal processo init, ma rimangono residenti in memoria in genere finché non ricevono un segnale di terminazione esplicito, e il buon programmatore cerca generalmente di evitare questo comportamento. Inoltre è indispensabile che il processo figlio ritorni esplicitamente un valore al padre tramite exit() quando è terminato, per evitare che quest'ultimo resti bloccato sulla wait in attesa che il figlio termini.

COMUNICAZIONE FRA PROCESSI - PIPE E SEGNALI

Dopo aver imparato come creare, gestire e terminare processi multipli, è indispensabile sapere come far comunicare fra loro questi processi. La via generalmente più semplice è quella del file. I processi possono scrivere su un file condiviso, che può essere un file comune o una FIFO creata via mkfifo(), le informazioni che desiderano scambiarsi. Un modo più raffinato può essere la syscall mmap(), che mappa un file in memoria e permette a due o più processi di accedere a quella locazione di memoria, mantenendo le scritture in memoria sincronizzate con quelle sul filesystem. Un'altra via può essere la primitiva pipe(), che crea una zona di memoria condivisa fra i processi dove possono essere inviate e lette informazioni, come se fosse un canale virtuale. pipe() in realtà non crea un solo canale bidirezionale, come farebbe socket(), ma due canali monodirezionali, uno per la lettura e uno per la scrittura, ognuno identificato dal proprio pipe descriptor. Per convenzione, il canale con indice 0 della pipe è usato per la lettura, quello con indice 1 per la scrittura. La scrittura sul canale di lettura, o viceversa la lettura sul canale di scrittura, comporta la terminazione anomala del programma con errore di broken pipe. Esempio:

```
int pp[2];
int pid;
char str[100];

/* Inizializzo la pipe */
if ( pipe(pp) < 0 ) { /* Errore */ }

pid = fork();

if ( pid == -1 ) { /* Errore */ }
else if ( pid == 0 ) {
        /*** Figlio ***/
        /* Chiudo il canale di lettura perché non mi serve */
        close ( pp[0] );

        /* Inizializzo e scrivo il messaggio sulla pipe.
           Tale messaggio verrà letto dal padre */
```

```
        strncpy ( str, sizeof(str), "Mes-
saggio dal figlio" );
        write ( pp[1], str, sizeof(str) );

        /* Chiudo la pipe ed esco */
        close ( pp[1] );
        exit ( 0 );
} else {
        /*** Padre ***/
        /* Chiudo il canale di scrittura */
        close ( pp[1] );

        /* Leggo il messaggio che il
figlio mi manda dalla pipe */
        read ( pp[0], str, sizeof(str) );
        printf ( "Il figlio ha inviato:
'%s'\n", str );

        /* Attendo che il figlio termini */
        while ( wait ((int*) 0 ) > 0 );
}
```

Tale meccanismo è previsto di default dalla maggior parte delle shell. Digitare il comando

```
ls -l | less
```

equivale a dire, a basso livello, "crea una pipe fra il processo associato al comando ls -l e quello associato al comando less. L'output del primo comando (che normalmente verrebbe inviato a stdout) diventa l'input della pipe che verrà letta da less". Ciò è possibile perché sotto Unix stdin e stdout sono due descrittori di file come tutti gli altri, quindi possono essere chiusi volendo (in tal caso il processo, rispettivamente, non potrà leggere nulla da stdin o scrivere nulla su stdout). Ma dopo averli chiusi è possibile effettuare l'operazione di duplicazione attraverso la primitiva dup(). Se faccio una cosa del genere

```
int pp[2];
pipe ( pp );

...

close ( 1 );
dup ( pp[1] );
```

chiudo il descrittore di stdout e duplico quest'ultimo sul descrittore di output della pipe (dup() duplica il descrittore passato come argomento sul primo descrittore libero che trova, in questo caso 1, ovvero stdout, se volessimo duplicarlo su un descrittore specifico dovremmo usare dup2()). Questa scrittura vuol dire che da quel punto in poi tutte le scritture effettuate su stdout verranno effettuate sulla pipe invece che sul terminale. Nel caso di ls -l | less, il processo associato a ls chiude il suo descrittore di stdout e manda sulla pipe le informazioni che normalmente invierebbe lì. Tali informazioni vengono lette dalla pipe dal processo less (la scrittura vista sopra, infatti, è perfettamente riproducibile per chiudere stdin e duplicare il descrittore sul canale di input di una pipe). Un'altra strategia comune per la comunicazione fra processi è lo scambio di segnali. Non sempre la comunicazione fra processi consiste nello scambio di informazioni: molte volte capita semplicemente che un processo debba attendere che un altro processo sia pronto o in un certo stato. Per fare questo si usano generalmente i segnali. I segnali vengono usati generalmente dal sistema operativo per terminare, sospendere o riprendere un processo, o anche per segnalare una terminazione anomala del processo stesso (segmentation fault, illegal instruction, broken pipe, floating point exception, e così via). Un processo può inviare un segnale a un altro processo attraverso la primitiva kill(), che prende come parametri il PID del processo a cui inviare il segnale e il codice del segnale da inviare (la lista delle macro associate ai codici dei segnali è visibile attraverso il comando man 7 signal). E attraverso la primitiva signal() si può personalizzare l'azione da eseguire quando un processo riceve un particolare segnale. Tale funzione prende come argomenti il codice del segnale da personalizzare e la funzione che dovrà essere richiamata quando quel segnale è ricevuto. In questo modo, ad esempio, se voglio intercettare il segnale SIGTERM inviato al mio processo, in modo da chiudere un descrittore di file prima che il processo venga terminato, posso scrivere un codice di questo tipo:

```
void term ( int sig )
{
        printf ( "Ricevuto il segnale %d -
Termino\n", sig );
        close ( fd );
        exit ( 1 );
}
...

/* Installo un handler per il segnale SIG-
TERM */
signal ( SIGTERM, term );
```

Dopo il seguente codice, la funzione term verrà richiamata quando il processo riceve il segnale SIGTERM. Attraverso questa politica si possono inibire alcuni segnali di terminazione sul processo (ad esempio il processo può ignorare un segnale SIGTERM o SIGINT, ma in nessun modo è però possibile inibire un segnale SIGKILL), o eseguire del codice particolare in caso di terminazione anomala (ad esempio in caso di SIGSEGV o SIGABRT). Ma la cosa più interessante è che attraverso questo meccanismo è possibile far comunicare i processi, ad esempio mettere in attesa un processo attraverso la funzione pause() fino a quando un altro processo non è pronto. Quando quest'ultimo è pronto, invierà al processo in attesa un segnale personalizzato (lo standard Unix mette a disposizione due segnali speciali, SIGUSR1 e SIGUSR2, che possono essere liberamente associati a funzioni arbitrarie senza il rischio di compromettere la stabilità del programma o del sistema effettuando l'override di segnali assegnati ad altri scopi). Se si vuole che la ricezione di SIGUSR* non faccia niente se non risvegliare un processo in pausa, si può associare al segnale una funzione vuota. Esempio:

```
void do_nothing ( int sig ) {}
...

/* Associo al segnale SIGUSR1 la funzio-
ne che non fa nulla
  e creo un nuovo processo figlio */
signal ( SIGUSR1, do_nothing );
int pid = fork();


/* Processo figlio */
...
pause();
printf ( "Mio padre e' pronto e mi ha
risvegliato\n" );
...

/* Processo padre */
/* ...codice... */
/* Risveglio il processo figlio mandandogli
un SIGUSR1 */
kill ( pid, SIGUSR1 );
```

PROBLEMI DI MUTUA ESCLUSIONE E SEMAFORI

Un problema difficile da gestire (e da debuggare) in software multiprocesso complessi è quello dell'accesso concorrente alle stesse locazioni di memoria. Diversi processi possono infatti accedere allo stesso momento alle stesse zone di memoria (variabili, strutture, array...). Se diversi processi accedono contemporaneamente in lettura alla stessa zona di memoria non ci dovrebbero essere grossi problemi, ma se i processi accedono contemporaneamente in scrittura il software può avere problemi di inconsistenza delle informazioni che possono portare anche al suo crash. Un processo A può andare, ad esempio, a leggere quanto denaro è presente su un conto corrente e, dopo diverse operazioni, può decidere di settare il valore del conto a quello precedente più il 10%. Se intanto il processo B effettua un versamento sul conto, ci si trova davanti a un grosso problema: A andrà a scrivere come valore del conto il valore letto precedentemente più il 10%, annullando la modifica effettuata intanto da B, e quindi rendendo la transazione di versamento nulla. Questo, come si può intuire, è un problema non indifferente quando si vanno a gestire diversi processi che scrivono sugli stessi dati. La soluzione implementata da sistemi Unix è quella dei semafori. Nel caso illustrato precedentemente, la soluzione al problema è la seguente:

- Il semaforo viene inizializzato (primitiva sem_init)
- Il processo A attende al semaforo (primitiva sem_wait), controllando che non ci sia nessun altro processo che stia usando quel semaforo
- Se non c'è nessun altro processo in esecuzione sul semaforo, A continua, effettuando le sue operazioni (lettura e aggiornamento del valore del conto corrente), e liberando il semaforo quando ha terminato di operare su quei dati (primitiva sem_post)
- Se il processo B vuole accedere a quei dati mentre A è in esecuzione, troverà al momento della sem_wait il semaforo occupato, e rimarrà in attesa fino a quando A non avrà terminato le sue operazioni. Al momento della sem_post, B riceve il via libera per operare sui dati del conto corrente, e questa volta il versamento viene effettuato correttamente senza perdita di dati. Un codice completo che utilizzi i semafori non può essere pubblicato in questa sede per motivi di spazio, ma grazie alle pagine di manuale di sem_init, sem_wait e sem_post (e sem_close per chiudere il semaforo quando non serve più), e anche i numerosi esempi online, è possibile senza grossi problemi scrivere un'applicazione multiprocesso dove i diversi task non risentano di problemi di accesso concorrente. Per utilizzare tale implementazione dei semafori è necessario includere l'header semaphore.h, e linkare l'eseguibile con l'opzione di gcc -lrt.

THREAD

Finora abbiamo parlato di programmazione multiprocesso. Un processo è spesso ben gestibile, in quanto completamente indipendente dal processo padre una volta avviato, ma relativamente oneroso dal punto di vista computazionale. Una volta avviato un processo figlio, infatti, il sistema operativo effettua nel nuovo process descriptor una copia della memoria del processo padre al momento della fork. Questo è relativamente oneroso in quanto richiede tempo (tempi di accesso e scrittura in memoria centrale) e spazio (se al momento della fork il processo padre occupa molta memoria, tutta questa memoria viene copiata all'interno del processo figlio). Non sempre si vuole questo, molto spesso si vuole che task separati siano avviati in tempi molto più rapidi, o che questi task possano condividere più facilmente la stessa memoria, senza che questa venga ricopiata ogni volta in una nuova copia indipendente da quella padre. Si parla in questo caso di thread, o lightweight processes (LWP). Il trend attuale è quello di preferire questi ultimi ai processi classici in un software multitasking, e anche progetti abbastanza grossi, come il server web Apache, negli ultimi tempi sono stati riscritti in un'ottica più multithread che multiprocesso. L'implementazione dei thread che esamineremo qui brevemente è quella POSIX, standard sui sistemi Unix-based, nota come POSIX thread o pthread. Per utilizzare tali funzioni è necessario includere l'header pthread.h, e linkare l'eseguibile con l'opzione di gcc -lpthread. Alla luce di quest'implementazione, un nuovo thread si crea attraverso la funzione pthread_create, che prende come parametri:

- Il riferimento a un thread descriptor (pthread_t).
- Il riferimento, opzionale, a un dato pthread_attr_t che contiene gli attributi del thread da creare.
- Una funzione che verrà richiamata al momento della creazione del thread.
- Il riferimento, opzionale, a una zona di memoria o variabile da passare come argomento al thread (trattata semplicemente come un puntatore a void).

Una volta avviato un thread, il processo che lo ha avviato può decidere di sincronizzarsi con quest'ultimo attendendo che termini e gli ritorni un valore, in modo simile a quello che fa la wait(), attraverso la funzione pthread_join, e il thread può ritornare un valore al chiamante e uscire via pthread_exit. Esempio:

```
#include <stdio.h>
#include <pthread.h>

void* thread ( void *arg )
{
        int *n = (int*) arg;
        printf ( "Sono un nuovo thread,
mio padre mi ha passato "
                        "il valore
%d\n", *n );
        pthread_exit ((void*) 0);
        return (void*) 0;
}
int main ( int argc, char **argv )
{
        pthread_t td;
        int *status;
        int n = 4;

        if ( pthread_create ( &td, NULL,
thread, &n ) != 0 )
        {
                printf ( "Creazione del
nuovo thread fallita\n" );
                return 1;
        }
        /* Attendo che il thread termini */
        pthread_join ( &pt, &status );
        printf ( "Il thread è terminato con
status %d\n", *status );
        return 0;
}
```